Enriching satellite image annotations of forests with keyphrases from a specialized corpus

Nathalie Neptune, Josiane Mothe

Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20015-2

1244: NEW METHODS OF AI AND DEEP LEARNING IN MULTIMEDIA

Enriching satellite image annotations of forests with keyphrases from a specialized corpus

Nathalie Neptune[1] · **Josiane Mothe**[2]

Received: 19 December 2023 / Revised: 6 June 2024 / Accepted: 2 August 2024

Abstract
The automatic annotation of changes in satellite images requires examples of appropriate annotations. Alternatively, keyphrases extracted from a specialized corpus can serve as candidates for image annotation models. In the case of detecting deforestation in satellite images, there is a rich scientific literature available on the topic that may serve as a corpus for finding candidate annotations. We propose a method that utilizes a deep learning technique for change detection and visual semantic embedding. This method is combined with an information retrieval framework to find annotations for pairs of satellite images showing forest changes. Our evaluation is based on a dataset of image pairs from the Amazon rainforest and shows that keyphrases provide richer semantic information without any negative impact on the annotation compared to annotating with single words.

Keywords Image annotation · Satellite image annotation · Deforestation annotation

1 Introduction

Deforestation is a major environmental problem that affects the global climate, biodiversity, and human well-being. Monitoring and understanding deforestation requires the integration of various data sources, such as satellite images and scientific publications. Satellite images provide spatial and temporal information on forest cover changes, while scientific publications provide contextual and explanatory information on the causes and consequences of deforestation. However, combining these data sources is challenging, as they have different formats, features, and content. In this paper, we propose a method that automatically

✉ Nathalie Neptune
nathalie.neptune@irit.fr

Josiane Mothe
josiane.mothe@irit.fr

1 IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Cr Rose Dieng-Kuntz, Toulouse 31400, State, France

2 IRIT,, Université de Toulouse, CNRS, Toulouse INP, UT3, INSPÉ, Cr Rose Dieng-Kuntz, Toulouse 31400, State, France

Published online: 13 August 2024

annotates satellite image pairs with relevant keyphrases from a specialized corpus of scientific publications.

We employ a visual semantic embedding model that learns to map image pairs to a high-dimensional semantic vector space, where they can be compared with word vectors. We use either pre-trained or corpus-trained word embeddings to obtain the vector representation of each keyphrase. We utilize a keyphrase extraction method to select the most relevant keyphrases from the corpus as candidate annotations. We then use an information retrieval framework to find the keyphrases most similar to the vector of the image pair. These keyphrases are the predicted annotations.

We evaluate our approach on a dataset of image pairs from the Amazon rainforest, and a corpus of scientific publications related to deforestation in the Amazon. The images were captured by the Landsat 8 Operational Land Imager[1], and the publications were collected from the Web of Science[2]. We compare our results with single keywords and different word embedding models. In terms of recall, our findings show that keyphrases reach the same performance as the single word annotations. Furthermore, the multi-word keyphrases that are proposed by the models provide additional semantic information that we might not get from single words alone. Additionally, we show how our approach can be used with different word embedding models including the state-of-the-art image-text embedding model CLIP [1].

Our contributions are as follows:

- **Annotation of Image Pairs:** We propose a method for annotating pairs of satellite images of forests, in the case of deforestation, in order to achieve higher recall compared to annotating single images [2].
- **Keyphrase Extraction:** We implement a technique to extract keyphrases from a specialized corpus related to the images, to use as candidate annotations, improving the quality of predicted annotations.
- **Embedding Model Comparison:** We evaluate the performance of different word embedding models (BERT, fastText, CLIP) on the annotation retrieval task, identifying fastText as the most effective overall.
- **Candidate Keyphrase Construction:** We analyze the impact of using n-grams and a grammar rule on extracting keyphrases from the corpus, showing that the grammar rule ensures semantically valid keyphrases.

We evaluate the performance of our approach using two specialized corpora from which we extract the candidate keyphrases.

2 Related work

The problem of adding semantic information to Earth observation (EO) images is well-studied in remote sensing and computer vision. In particular, many studies have focused on the particular task of semantic segmentation of satellite images which consists in labelling each pixel of the image with its class. Some approaches integrate ontologies into the segmentation process [3, 4]. Other approaches use geo-referenced Wikipedia articles that are matched with their corresponding satellite images [5]. The ontology-based approaches have been shown to improve classification accuracy, but they require significant expert input in building

[1] https://www.usgs.gov/landsat-missions/landsat-8

[2] https://www.webofscience.com/

the ontology. The Wikipedia-based approach resulted in only modest improvements in the semantic segmentation task, they do have the benefit of not requiring the use of experts.

Unlabelled satellite images are plentiful, one promising solution to extract semantic information from them is by using visual semantic embedding (VSE) models. These models use deep neural networks to learn representations of images and text in a common space. This common representation makes tasks such as image-text retrieval, image captioning and other image-text matching. Notably, the CLIP-RSICD[3] model, a large-scale transformer-based model [1], has been pre-trained on a dataset of satellite images and their captions [6], enabling it to perform zero-shot and fine-tuned annotation tasks. CLIP-RSICD and similar models can effectively capture visual concepts and generalize to unseen classes. However, it is worth noting that these models may struggle with providing relevant image annotations in a specific domain when faced with a large number of candidate annotations, as the likelihood of incorrect annotations increases.

Convolutional neural network (CNN) models have been shown to be effective at capturing features of satellite images and performing various specific tasks such as image segmentation for change detection in particular [2, 7–10]. The effectiveness of these models in learning annotations for satellite image pairs has been demonstrated, for forest images in previous work [2, 11, 12]. It has been shown that using image pairs as input rather than individual images, resulted in higher recall of correct single-word annotations.

Building upon this work, we now propose a novel approach that combines corpus keyphrase extraction with a text retrieval framework to annotate image pairs of forests, in a context where there are changes to be detected in the images. The goal is to improve annotation results by restricting candidate annotations to a subset of keyphrases that are highly relevant to the images. Extracting the keyphrases from a corpus that is specialized in the type of changes that can be observed in the images, allows us to find annotations that are relevant to the context of our images. We use two keyphrase extraction methods: PatternRank [13], and a method that uses n-grams as candidates with word embeddings from sentence-BERT [14], fastText [15] and CLIP [1]. These two methods leverage pre-trained large language models for unsupervised keyphrase extraction based on word embeddings. PatternRank [13] has been shown to outperform other state-of-the art unsupervised keyphrase extraction methods [16–18], on a corpus of scientific publications [19]. The other n-gram model that we use has shown good results in previous work specifically for annotating image pairs of forests [2, 11, 12].

Sentence-BERT is a modified pre-trained BERT model that uses siamese and triplet network structures to learn sentence embeddings that are semantically meaningful. PatternRank uses a complex part-of-speech pattern to select candidate keyphrases from a document. It then ranks the keyphrases based on their semantic similarity to the document using their embeddings from a sentence-BERT model. Our n-gram-based method selects the most common n-grams (1 to 3-grams) as candidate annotations and ranks them in the same way as PatternRank. FastText embeddings are word representations that incorporate subword information (character n-grams). FastText models are trained using either the continuous bag-of-words (CBOW) or the skipgram model [20]. In the CBOW model, a central word is surrounded by context words, and the model identifies the central word given its context. In the skipgram model, context words are predicted given a word or subword. fastText models can handle out-of-vocabulary words by summing up the vectors of their n-grams. CLIP embeddings represent images and text in a common vector space. The CLIP model is trained with a

[3] https://github.com/arampacha/CLIP-rsicd

contrastive objective to predict which image and text pairs are matched in a large dataset. CLIP embeddings enable many zero-shot tasks such as image classification and captioning.

We evaluate our approach on a multimodal dataset of satellite image pairs and scientific texts. We compare the performance of our models with CLIP-RSICD in zero-shot and after fine-tuning. Our approach is particularly suited for cases where the dataset contains image pairs with image-wise annotations made of words from a specialized domain. Moreover, our approach can be adapted to different types of EO images by modifying the network architecture accordingly. This flexibility allows for the application of our method to various domains outside of the environmental sciences, as long as image pairs with relevant document corpora are available.

3 Methods

Our goal is to learn annotations for pairs of satellite images by using keyphrases extracted from a specialized corpus as our candidate annotations. Formally, let $\mathcal{X}_{ij}^{t_1}$ and $\mathcal{X}_{ij}^{t_2}$ be two images of the same area, of $i \times j$ pixels, taken at different times t_1 and t_2, respectively. We want to annotate these images with keyphrases that reflect the content of the image and the type of change between the images if present. Our approach consists of three main steps. The first step is to learn the visual semantic embeddings of image pairs and their labels in the same embedding space. The second step is to extract the most important keyphrases from a specialized corpus related to the images using a state of the art keyphrase extraction method. The third and final step is to predict the annotations for test images by performing information retrieval with the image pair as the query and the most relevant annotations from the corpus as the results. Figure 1 shows an illustration of our approach.

3.1 Learning visual semantic embeddings for annotations

Let I be the pair of the images $\mathcal{X}_{ij}^{t_1}$ and $\mathcal{X}_{ij}^{t_2}$ that have been stacked, the resulting dimensions $i \times j \times 2C$, where C is the number of channels of each image. We want to learn a function

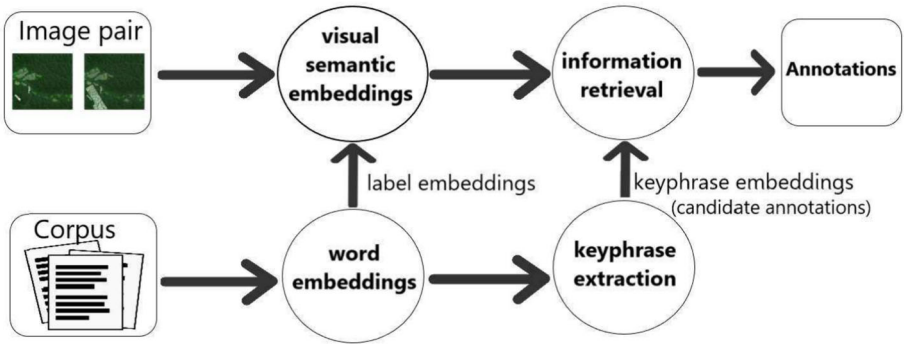


Fig. 1 Overview of our method. The components: visual semantic learning, keyphrase extraction from a specialized corpus and information retrieval for annotating satellite image pairs. An embedding of the image pair is learned to match the embedding of its label. Candidate embeddings are extracted from the corpus and provided as candidate annotations. Annotations are selected by the retrieval of the candidates that are most similar to the embedding of the image pair

$\mathcal{F}_\theta$ that maps I to a high-dimensional semantic vector $\mathcal{Z}_{ij} = f_\theta(I) \in \mathbb{R}^k$. $f(\cdot)$ is the learned embedding function, and θ is the set of learnable parameters of the neural network. Given $\mathcal{Z}_{ij}$, our objective is to retrieve the top n keyphrases $v_1, v_2, \ldots, v_n$ from the specialized corpus that are the most relevant to the content of the image pair. We also want to retrieve other relevant semantic information.

We use a residual network encoder (ResNet) [21] to encode the visual information from the images into the semantic vector. ResNet is a deep neural network architecture that improves on plain networks by adding residual learning to avoid errors that come with the increased depth of the networks. This improvement is achieved by using shortcut connections that skip one or more layers and perform identity mapping. ResNets are able to converge and reach state-of-the-art results on benchmark datasets even when the models are very deep (up to 152 layers). ResNet encoders have been shown to perform well as the encoding part of change detection networks [7, 8] including in the case of changes in forests [2].

The ResNet architecture that we utilize, to learn VSEs, is made of 34 layers (ResNet34). It was initially designed to be used for change detection in image pairs, as part of a segmentation model [2].

The architecture of the ResNet34 is as follows. The input layer is a 6-channel convolutional layer using a 7x7 convolution with 64 filters, a stride of 2, and padding of 3, described by $\text{Conv}(x) = W * x + b$, where x where W is the weight matrix, $*$ denotes the convolution operation, x is the input, and b is the bias term. Then batch normalization is applied $\text{BN}(x) = \gamma \left(\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} \right) + \beta$ where μ and σ^2 are the mean and variance of the batch, ϵ is a small constant for numerical stability, and γ and β are learnable parameters. This is followed by ReLU activation $\text{ReLU}(x) = \max(0, x)$) and a 3x3 max pooling layern $\text{MaxPool}(x) = \max(x_i)$ where x_i are the values within the pooling window, with a stride of 2 and padding of 1.

A series of residual blocks comprise the encoder. Each residual block has two 3x3 convolutional layers with batch normalization, ReLU activation, and shortcut connection. The function within a residual block is: $\text{F}(x) = \text{BN}(\text{Conv2}(\text{ReLU}(\text{BN}(\text{Conv1}(x)))))$. The operation carried out by residual block is: $\text{ResidualBlock}(x) = \text{ReLU}(x + \text{F}(x))$, where x is the input to the residual block and $\text{F}(x)$ is the output of the two convolutional layers within the block.

The ResNet34 architecture includes four stages of residual blocks:

- 1: 3 blocks with 64 filters.
- 2: 4 blocks with 128 filters.
- 3: 6 blocks with 256 filters.
- 4: 3 blocks with 512 filters.

For stages 2 to 4 the first block performs downsampling with a stride of 2. This configuration extracts multi-scale features that are subsequently fed into a regression head for generating the VSEs.

The elements of the regression head are a sequence of layers that process the features extracted by the ResNet34 encoder and generate regression outputs. First, the features are flattened into a one-dimensional vector, $x_{\text{flattened}}$. Then, this flattened representation goes through dropout $\text{Dropout}(x_{\text{flattened}})$, linear transformation layers $\text{Linear}(x_{\text{dropout}}) = W_1 \cdot x_{\text{dropout}} + b_1$, ReLU activation $\text{ReLU}(x_{\text{linear}})$, and batch normalization $\text{BatchNorm}(x_{\text{relu}})$. After which, the features pass through another dropout layer $\text{Dropout}(x_{\text{normalized}})$, before going through a final linear layer $\text{Linear}(x_{\text{dropout2}}) = W_2 \cdot x_{\text{dropout2}} + b_2$.

Random initialization is used for the network weights. The dropout rate used is 0.25. For optimization, the Adam optimizer with a weight decay of 1×10^{-3} is used. A learning rate scheduler is applied to exponentially decay the learning rate by a factor of 0.95. The loss

function is cosine similarity, implemented as $1-\text{cosine\_similarity}(y_{\text{pr}}, y_{\text{gt}})$, where y_{pr} and y_{gt} are the predicted and ground truth values, respectively. The metric used for model selection is mean squared error (MSE), calculated as the sum of squared differences between predicted and ground truth values, normalized by the number of elements. The training process spans 40 epochs. The same hyperparameters are used for all the trained models.

The input of the encoder is an image pair and the target is the embedding of the label. Each image pair has a single label that is either "deforestation" or "forest" depending on whether there are pixels showing deforestation from the first image to the other or not.

For the label embeddings, we use pre-trained word embeddings, which can improve performance on downstream natural language processing tasks by transferring the extensive linguistic knowledge from the vast amounts of text data they are trained on. We test fastText embeddings trained on Common Crawl[4] and Wikipedia[5] with dimension 300 and a context size of 5 words. We also use sentence-BERT embeddings with dimension 768 and context size of 128 tokens trained on various web data sources. We use the BERT model xlm-r-bert-base-nli-stsb-mean-tokens from sentence-BERT. In addition, we use a fastText model that we train on the Forest corpus and we refer to this embedding model as "fastText custom."

3.2 Keyphrase extraction from a specialized corpus and annotation retrieval

We select a corpus of scientific publications that are related to our images in terms of the changes that might be observed and the area of interest. With keyphrase extraction, we want to select a set of candidate annotations for our images pairs from this specialized corpus. For each document we extract its most important keyphrases using a method similar to keyBERT [18] where keyphrases are extracted based on the cosine similarity of their embeddings with the embedding of the document.

Let $\mathcal{G}$ be the corpus of m documents, $\mathcal{G} = d_1, d_2, \ldots, d_m$, where each document d_i contains information relevant to the type of change present in the images, and potentially the specific geographical region. For each document $d_i \in \mathcal{G}$, we compute the document embedding. We then evaluate the similarity between each document embedding and the embeddings of the keyphrases present within the document.

We use three word embedding models for extracting keyphrases: BERT, fastText and CLIP-RSICD. In each case, we take the same model used for the embeddings of the labels of the images to get the embeddings of the documents and keyphrases. For n-gram keyphrases, we take all n-grams with n having a maximum value of 3. The n-grams are then ranked from most similar to least similar to the document base on cosine similarity.

We also use PatternRank to pre-select keyphrases as an alternative to selecting them based on n-grams. When using PatternRank, the candidate keyphrases are first pre-selected based on a grammar rule: ((.*HYPH.*) NOUN*) || $\left((\text{VBG}|\text{VBN})?\text{ADJ}^*\text{NOUN}^+\right)$ [13].

This grammar rule matches specific noun phrase patterns in the text. It captures either noun phrases containing hyphens, or sequences of one or more nouns that may be preceded by adjectives, and certain verb forms (gerunds or past participles). This rule ensures that the candidates are semantically valid keyphrases. Then these pre-selected keyphrases are ranked based on the cosine similarity of their BERT embeddings with the embedding of the document.

[4] https://commoncrawl.org/ - Common Crawl.

[5] https://www.wikipedia.org/

After extracting keyphrases from all documents in the corpus, we aggregate them and order them based on their frequency across the entire corpus. These keyphrases are the candidate annotations for our image pairs.

We then compare the semantic vector $\mathcal{Z}_{ij}$ with the embedding of the candidate annotations based on their cosine similarity and retrieve the most similar candidates. The semantic vector is therefore used as a query to retrieve the annotations from the candidate keyphrases extracted from the corpus. These keyphrases are returned ordered from most similar to least similar, and they are the annotations predicted by our method. A post-processing step is used to keep either the plural or the singular form of a keyphrase, whichever appears first in the list of predicted keyphrases. We do not remove synonyms because we are aiming for exact-match keyphrases and therefore are not counting synonyms as matches, neither are we counting partial matches.

3.3 Evaluation

We evaluate the performance of the annotation task by using the recall at k (R@k) similarly to [2, 22, 23]. R@k measures the recall of the target annotation among the top k predicted annotations. In our setting, the image pair is the query and the results are the top k annotations that were found to be the most similar to the image pair. For each image pair, R@k is equal to 1 if the target is present in the top k annotations predicted by the model and 0 it not. We report the average value of R@k for all the image pairs tested for k values of 1, 5 and 10. We use the CLIP-RSICD model as a baseline to evaluate our approach.

4 Experiments and results

4.1 Datasets

4.1.1 Satellite images

We use images of the Landsat scene 230_65 for dates June 21, 2017, June 24, 2018, and July 13, 2019. This dataset covers an area of interest in the Brazilian Amazon that has been affected by deforestation during the time period selected, and was first introduced by [10] and used for bi-temporal change detection. The images provided by the United States Geological Survey are Landsat 8 tier 1 imagery, they are high quality, analysis-ready and suitable for time series analysis. We process the images by selecting bands 4, 3 and 2 corresponding to Red, Green and Blue color bands, and we create non-overlapping tiles of 256 x 256 pixels, for each image. Images are tiled before passing through the network, as is common in the literature, in particular for image segmentation and change detection in satellite images [2, 7, 9, 10]. Using batches of tiles instead of whole images at once allows to preserve memory, it also helps improve performance. A 75%-25% split is used to divide the training and testing tiles that are selected randomly, resulting in 319 image pairs for training and 80 for testing per year. There is no spatial overlap between the training and the testing dataset. The pixel labels used to derive the image labels are available from the data producers [10] with data from the Brazilian Institute of Space Research's Project for Deforestation Mapping [24]. The image pairs showing deforestation are labelled "deforestation" and the image pairs not showing deforestation are labelled "forest." The dataset that we are using is relatively small, we perform data augmentation including horizontal flip with a probability of 50%, a

combination of scaling, shifting, and rotating the image with specific limits. The image can be scaled up or down by up to 50%, shifted horizontally or vertically by up to 10%, and rotated with no rotation limit, all with a probability of 100%.

4.1.2 Corpora

To ensure that we have text that is specifically related to deforestation in the Amazon during the years 2018 to 2019, we built a corpus with publications from the Web of Science[6] using the keywords "Amazon Brazil deforestation" taking only publications published from 2017 to 2020. This corpus contains 446 publications and is referred to as the "Amazon corpus." Additionally, we use another corpus, referred to as the "Forest corpus" [25] containing 9722 publications obtained form the Web of Science using the keyword "deforest*" spanning the years 1975 to 2016, this corpus covers the topic of deforestation in a broader context. These corpora are used to extract candidate keywords for annotating the images pairs. We also use the "Forest corpus" to train the word embedding model called "fastText custom."

Corpus pre-processing involves removing stopwords, using the NLTK[7] English stopword list as well as characters that are not letters and words with fewer than three characters.

4.2 Experimental setup

The version of CLIP-RSICD that we use is the flax-community/clip-rsicd-v2[8] from OpenAI. For zero-shot and fine-tuning on our own data as a baseline to compare to our proposed models. In zero-shot, we take the embeddings of the second image in the pair obtained and compare with the embeddings of the candidates annotations by the same model, with cosine similarity. Candidate annotations were extracted using the keyphrase extraction approach described in Section 3.2.

For the fine-tuning process, we load the data with a batch size of 8 with shuffling enabled. We use the AdamW optimizer with a learning rate of 5×10^{-5}, betas of (0.9, 0.98), epsilon of 1×10^{-6}, and a weight decay of 0.2. The contrastive loss function used is defined as:

$$L = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(d_i)^2, & \text{if } d = 0 \\ \frac{1}{n}\sum_{i=1}^{n}[\max(0, m - d_i)]^2, & \text{if } d = 1 \end{cases}$$

Where n is the batch size, d_i is the Euclidean distance between the corresponding pair of embeddings $y1$ and $y2$, m is the margin or radius parameter, $d = 0$ means the pair of embeddings should be similar (positive pairs), $d = 1$ means the pair of embeddings should be dissimilar (negative pairs). This function is used with a margin of 2.0. The training was carried out for 30 epochs. One GPU device, an Nvidia GTX1080Ti was used for all experiments.

4.3 Evaluation of results

We do the evaluation using the recall values at 1, 5 and 10. We also perform a qualitative analysis, comparing the keyphrases selected as annotations by each model. We compare the performance of our models with CLIP-RSICD. Table 1 shows the values of recall at 1, 5 and 10 for all the VSE models and the different keyphrase extraction methods used.

[6] https://www.webofscience.com/

[7] https://www.nltk.org/

[8] https://huggingface.co/flax-community/clip-rsicd-v2

Table 1 Average recall at 1, 5 and 10 with 25 and 100 candidates

VSE model	Keyphrase extraction method	R@1	R@5	R@10
25 candidates				
BERT	BERT keywords	0.65	0.99	0.99
BERT	BERT n-gram keyphrases	0.65	0.99	0.99
BERT	PatternRank keyphrases 1	0.65	0.99	0.99
BERT	PatternRank keyphrases 2 (Forest corpus)	0.65	0.99	0.99
fastText	fastText keywords	0.70	0.94	0.94
fastText	fastText keyphrases	0.70	0.94	0.94
fastText	PatternRank keyphrases 1	0.70	0.94	0.94
fastText	PatternRank keyphrases 2 (Forest corpus)	0.70	0.94	0.94
fastText custom	fastText custom keywords (Forest corpus)	0.70	0.95	0.96
fastText custom	fastText custom keyphrases (Forest corpus)	0.70	0.95	0.96
fastText custom	PatternRank keyphrases 1	0.70	0.95	0.96
fastText custom	PatternRank keyphrases 2 (forest corpus)	0.70	0.95	0.96
CLIP-RSICD	CLIP-RSICD keywords	0.00	0.55	0.96
CLIP-RSICD	CLIP-RSICD n-gram keyphrases	0.00	0.11	0.30
CLIP-RSICD	PatternRank keyphrases 1	0.00	0.28	0.90
CLIP-RSICD	PatternRank keyphrases 2 (Forest corpus)	0.00	0.23	0.89
CLIP-RSICD	CLIP-RSICD keywords (s)	0.12	0.12	0.12
CLIP-RSICD	CLIP-RSICD n-gram keyphrases (s)	0.00	0.00	0.00
CLIP-RSICD	CLIP-RSICD PatternRank keyphrases 1 (s)	0.24	0.24	0.24
CLIP-RSICD	CLIP-RSICD PatternRank keyphrases 2 (Forest corpus) (s)	0.09	0.09	0.09
100 candidates				
BERT	BERT keywords	0.65	0.99	0.99
BERT	BERT n-gram keyphrases	0.50	0.82	0.90
BERT	PatternRank keyphrases 1	0.59	0.84	0.96
BERT	PatternRank keyphrases 2 (Forest corpus)	0.59	0.69	0.86
fastText	fastText keywords	0.70	0.94	0.94
fastText	fastText keyphrases	0.70	0.94	0.94
fastText	PatternRank keyphrases 1	0.70	0.82	0.93
fastText	PatternRank keyphrases 2 (Forest corpus)	0.70	0.82	0.93
fastText custom	fastText custom keywords (Forest corpus)	0.70	0.95	0.96
fastText custom	fastText custom keyphrases (Forest corpus)	0.70	0.95	0.96
fastText custom	PatternRank keyphrases 1	0.70	0.78	0.94
fastText custom	PatternRank keyphrases 2 (forest corpus)	0.70	0.74	0.94

fastText models outperform BERT models in terms of recall, especially when the number of candidates is high. The recall is lower when the number of candidates is higher for most models except for the fastText models with very few exceptions. Using the grammar rule of PatternRank to select the candidate keyphrases does not change when 25 candidates are used (except for CLIP-RSICD) while providing more semantic information than single keywords. (s) indicates that the keyphrases are used within a sentence

When the VSE model is trained with fastText embeddings, and the candidate keyphrase annotations are extracted with fastText, recall is higher compared to training the VSE model with BERT embeddings and extracting candidates with BERT. The fastText models proved to not be as negatively impacted by a large number of candidates.

In Table 1, we see that for recall at 1, fastText models outperform BERT models. The performance gap between the two is not very big only 0.05 for recall at 1 with 25 candidates. In fact, all BERT models reach 0.99 recall at 5 while the fastText models only reach at 0.94 and 0.94. Yet, using a fastText model for this task given our experimental setting is more efficient because fastText models are smaller and require less memory than BERT.

We report results with 100 annotation candidates, also in Table 1, to see if a large number of candidates makes the models more error prone. For fastText the recall at 1 remains the same, similarly for the BERT model using keywords. However, with keyphrase candidates the BERT models have a lower recall on average. BERT models are more sensitive to the high number of candidates. Therefore, if BERT models are chosen, care should be taken to select only the best possible candidates to ensure the best performance.

We tested with CLIP-RSICD with zero-shot prediction of keywords and keyphrases, then we tested with the same keywords and keyphrases integrated into a sentence. We used the sentences "a satellite image showing a forest" and "a satellite image showing deforestation" for each label, "forest" and "deforestation" respectively. After testing a few other variations we found that this sentence gives the best results in terms of recall. The sentence structure improved the results for CLIP-RSICD for recall at 1 only from 0.00 to 0.12 as seen in Table 1. We also tested with CLIP-RSICD finetuned on our dataset and found that the finetuning did not change the results, the recall values remained the same after finetuning. Overall, all our models outperform CLIP-RSICD. With 100 candidates, the recall for CLIP-RSICD is 0.00 on average (not reported in Table 1).

Table 2 shows the values of recall at 1 for the true label (R@1) compared to the recall at 1 of the other label (oR@1). If the true label is 'forest', the other label is 'deforestation' and vice versa. The difference between the two values is also reported to highlight the discriminative power of the models. The fastText models have the highest values for R@1 and also lowest values for oR@1. These models are therefore less likely to predict the other label in the top annotations.

The results show that compared to using single keywords as candidate annotations, keyphrases do not have a significant negative impact on recall at 1, for up to 100 candidate annotations. This shows that the top keyphrases are indeed keywords, matching the labels of the image pairs in our dataset.

For qualitative evaluation, we show two sample image pairs from the test dataset, the first sample shows deforestation while the second shows a full forest cover. This two samples illustrate the two labels present in our dataset and allow us to see the output of the models including in cases where they make wrong predictions. Tables 3, 4, 5 and 6 show the top annotations for the different methods along with their cosine similarities with the shown image pair.

In Table 3, we see the top 5 annotations obtained with the BERT models. All the models correctly predict the label 'deforestation' as the top 1 annotation. With 25 candidates, all the models also predict 'forest' as the second annotation. While this might be visually correct (the image showing deforestation within an forest) we would expect more annotations related to the label. With more candiate annotations (we show the case of 100) only 1 out of 4 models predicts the second annotation as 'forest', the others predict 'deforestation' and 'deforestation rates'. The keyphrase-based BERT (simply called BERT) incorrectly predicts 'deforestation forest' as the top 1 annotation. This ngram is not semantically valid. With the

Table 2 Difference between true label recall at 1 (R@1) and other label recall at 1 (oR@1) for all models with up to 1000 candidate annoations

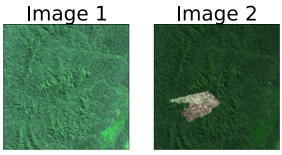
VSE Model	Keyphrase extraction method	R@1	oR@1	diff.
25 candidates				
BERT	BERT keywords	0.65	0.34	0.31
BERT	BERT n-gram keyphrases	0.65	0.34	0.31
BERT	PatternRank keyphrases 1	0.65	0.34	0.31
BERT	PatternRank keyphrases 2 (Forest corpus)	0.65	0.34	0.31
fastText	fastText keywords	0.70	0.24	0.46
fastText	fastText keyphrases	0.70	0.24	0.46
fastText	PatternRank keyphrases 1	0.70	0.24	0.46
fastText	PatternRank keyphrases 2 (Forest corpus)	0.70	0.24	0.46
fastText custom	fastText custom keywords (Forest corpus)	0.70	0.26	0.44
fastText custom	fastText custom keyphrases (Forest corpus)	0.70	0.26	0.44
fastText custom	PatternRank keyphrases 1	0.70	0.26	0.44
fastText custom	PatternRank keyphrases 2 (Forest corpus)	0.70	0.26	0.44
100 candidates				
BERT	BERT keywords	0.65	0.34	0.31
BERT	BERT n-gram keyphrases	0.50	0.14	0.36
BERT	PatternRank keyphrases 1	0.59	0.24	0.35
BERT	PatternRank keyphrases 2 (Forest corpus)	0.59	0.24	0.35
fastText	fastText keywords	0.70	0.24	0.46
fastText	fastText keyphrases	0.70	0.24	0.46
fastText	PatternRank keyphrases 1	0.70	0.24	0.46
fastText	PatternRank keyphrases 2 (Forest corpus)	0.70	0.24	0.46
fastText custom	fastText custom keywords (Forest corpus)	0.70	0.26	0.44
fastText custom	fastText custom keyphrases (Forest corpus)	0.70	0.26	0.44
fastText custom	PatternRank keyphrases 1	0.70	0.26	0.44
fastText custom	PatternRank keyphrases 2 (Forest corpus)	0.70	0.26	0.44
1000 candidates				
BERT	BERT keywords	0.51	0.26	0.25
BERT	BERT n-gram keyphrases	0.41	0.12	0.29
BERT	PatternRank keyphrases 1	0.38	0.12	0.26
BERT	PatternRank keyphrases 2 (Forest corpus)	0.55	0.19	0.36
fastText	fastText keywords	0.70	0.24	0.46
fastText	fastText keyphrases	0.61	0.09	0.52
fastText	PatternRank keyphrases 1	0.61	0.09	0.52
fastText	PatternRank keyphrases 2 (Forest corpus)	0.70	0.24	0.46
fastText custom	fastText custom keywords (Forest corpus)	0.70	0.26	0.44
fastText custom	fastText custom keyphrases (Forest corpus)	0.64	0.12	0.52

Table 2 continued

VSE Model	Keyphrase extraction method	R@1	oR@1	diff.
fastText custom	PatternRank keyphrases 1	0.64	0.12	0.52
fastText custom	PatternRank keyphrases 2 (Forest corpus)	0.70	0.26	0.44

The other label recall at one (oR@1) is the recall of the other label from our image dataset. When the true label of the image pair is "forest", the other label is "deforestation" and vice versa. When the number of candidate annotations is very large (1000 shown here), most models show lower recall at 1 values. The BERT models have the lowest difference between R@1 and oR@1. This means that these models are more likely to get the top 1 annotation wrong. The fastText models show the highest differences between R@1 and oR@1, in particular the models pre-trained on Common Crawl and Wikipedia, which slightly outperform our fastText custom models

Table 3 Sample image pair showing deforestation and the top 5 predicted annotations from 25 and 100 candidate keyphrases extracted with different keyphrase extraction methods and the BERT VSE model

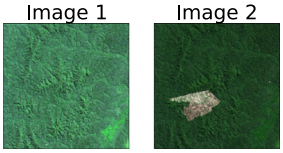


Label: deforestation

Model	Top 5 annotations	Cosine Similarities
25 candidates		
BERT keywords	deforestation, forest, vegetation, environmental, conservation	0.97, 0.69, 0.54, 0.50, 0.49
BERT	deforestation, forest, vegetation, environmental, *land use*	0.97, 0.69, 0.54, 0.50, 0.50
PatternRank 1	deforestation, forest, vegetation, degradation, *land cover*	0.97, 0.69, 0.54, 0.52, 0.51
PatternRank 2 (Forest corpus)	deforestation, forest, vegetation, degradation, *land cover*	0.97, 0.69, 0.54, 0.52, 0.51
100 candidates		
BERT Keywords	deforestation, forest, rainforest, deforested, biomass	0.97, 0.69, 0.67, 0.61, 0.58
BERT	*deforestation forest*, deforestation, *deforestation rates*, *forest degradation*, *deforestation amazon*	0.98, 0.97, 0.93, 0.85, 0.76
PatternRank 1	deforestation, *deforestation rates*, *forest degradation*, *forest loss*, *amazon deforestation*	0.97, 0.93, 0.85, 0.74, 0.73
PatternRank 2 (Forest corpus)	deforestation, *deforestation rates*, *forest degradation*, *tropical deforestation*, reforestation	0.97, 0.93, 0.85, 0.84, 0.82

The label of this image pair is 'deforestation' and all but one of the models shown correctly predict it as the top 1 annotation. When given 100 candidates (instead of 25) the keyphrase-based models are less likely to return 'forest' as a top annotation. The models are able to find more annotations with higher cosine similarity values among keyphrase candidates compared to keywords-only candidates

Table 4 Sample image pair showing deforestation and the top 5 predicted annotations from 25 and 100 candidate keyphrases extracted with different keyphrase extraction methods and the fastText VSE model



Label: deforestation

Model	Top 5 annotations	Cosine similarities
25 candidates		
fastText keywords	forest, deforestation, land, areas, change	0.98, 0.72, 0.50, 0.45, 0.45
fastText	forest, deforestation, *land use*, land, areas	0.98, 0.72, 0.51, 0.50, 0.45
fastText PatternRank 1	forest, deforestation, *land cover*, degradation, *land use*	0.98, 0.72, 0.54, 0.53, 0.51
fastText PatternRank 2 (Forest corpus)	forest, deforestation, *land cover*, degradation, *land use*	0.98, 0.72, 0.54, 0.53, 0.51
100 candidates		
fastText Keywords	forest, deforestation, degradation, land	0.98, 0.72, 0.53, 0.50, 0.45
fastText	forest, deforestation, *land cover*, degradation, *land use*	0.98, 0.72, 0.54, 0.53, 0.51
fastText PatternRank 1	forest, *forest degradation*, *forest loss*, *forest cover*, *amazon forest*	0.98, 0.89, 0.85, 0.84, 0.80
fastText PatternRank 2 (Forest corpus)	forest, *forest degradation*, *forest loss*, *tropical forest*, *forest cover*	0.98, 0.89, 0.85, 0.84, 0.84

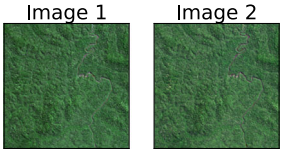
All the models shown incorrectly predict 'forest' as top 1 annotation for this image pair. Six out of eight models return 'deforestation' as the second annotation and the remaining two return the keyphrase 'forest degradation' which is closely related to deforestation

grammar rule applied, the PatternRank models predict only semantically valid keyphrases such as 'deforestation rate', 'forest degradation', 'forest loss' and 'tropical deforestation' and 'amazon deforestation', which are all related to deforestation in the Amazon as seen in the sample image pair.

Table 4 shows the top 5 keywords that were obtained with fastText models, for the same deforestation image pair. These fastText models were pre-trained on Common Crawl and Wikipedia. None of the models was able to correctly predict 'deforestation' as the top 1 annotation. With 25 candidates, they succeed in predicting it as the second annotation. With 100 candidates, only the keyword and ngram-based models predict it as the second annotation. The PatternRank models, with 100 candidates, predict 'forest degradation' and 'forest loss' as second and third annotations. Both keyphrases are directly related to deforestation.

In Table 5 the image pair is showing a full forest. All models shown are BERT-based models and they all correctly predict 'forest' as the top 1 annotation. With 25 candidates, the keyword-based model and the ngram-based model predict 'deforestation' as the fifth annotation, this is due to the fact that they produce keyphrases and keywords with higher cosine similarities than 'deforestation'. The third annotation with the keyword model are

Table 5 Sample image pair showing the forest and the top 5 predicted annotations from 25 and 100 candidate keyphrases extracted with different keyphrase extraction methods and the BERT VSE model



Label: Forest

Model	Top 5 annotations	Cosine similarities
25 candidates		
BERT Keywords	forest, vegetation, cover, land, deforestation	1.00, 0.66, 0.51, 0.50, 0.49
BERT	forest, vegetation, *land use*, land, deforestation	1.00, 0.66, 0.51, 0.50, 0.49
PatternRank 1	forest, vegetation, *land cover*, ecosystem, pasture	1.00, 0.66, 0.58, 0.52, 0.51
PatternRank 2 (Forest corpus)	forest, vegetation, landscape, *land cover*, ecosystem	1.00, 0.66, 0.60, 0.58, 0.52
100 candidates		
BERT Keywords	forest, rainforest, vegetation, landscape, rural	1.00, 0.76, 0.66, 0.60, 0.60
BERT	forest, forest cover, *forest degradation*, rainforest, *tropical forests*	1.00, 0.95, 0.79, 0.76, 0.67
PatternRank 1	forest, *forest cover*, *forest code*, *forest degradation*, *forest loss*	1.00, 0.95, 0.83, 0.79, 0.77
PatternRank 2 (Forest corpus)	forest, *forest area*, *forest cover*, forestry, *forest management*	1.00, 0.98, 0.95, 0.89, 0.87

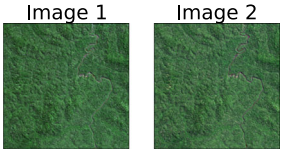
All the models shown correctly predict the top 1 annotation 'forest'. With 25 candidates, when we use keyphrases we obtain annotations such as 'land use' and 'land cover' compared to the single keywords 'land' and 'cover'. With 100 candidates we get more annotations with higher cosine similarity to the image pair

'cover'. Taken alone it does not provide a lot of information, compared to the PatternRank 1 model, which provides 'land cover' as its third annotation with higher cosine similarity than 'cover' and 'land' taken separately. With 100 candidates, the keyphrase-based models predict several annotations containing the word 'forest'. The annotations have high cosine similarities compared to others that do not contain the word 'forest'. Some of these annotations like 'tropical forests' are very precisely related to the image pair. Others, however are more related to deforestation such as 'forest degradation'. In this case, the cosine similarity seems to reflect more the composition of the keyphrase than its actual meaning.

Table 6 shows the top 5 annotations obtained with the fastText models for the same sample image showing a full forest. All models correctly predict the top 1 annotation as 'forest'. With 25 candidates they also all predict 'deforestation' as the second annotation.

Overall, when given only 25 candidates, the models tend to predict the other label as a second annotation. This phenomenon is mitigated when the models are given a higher number of candidate annotations. In general, with 100 candidates the models are able to retrieve annotations with higher cosine similarities compared to only having 25 candidates. More

Table 6 Sample image pair showing the forest and the top 5 predicted annotations from 25 ad 100 candidate keyphrases extracted with different keyphrase extraction methods and the fastText VSE model



Label: Forest

Model	Top 5 annotations	Cosine similarity
25 candidates		
fastText Keywords	forest, deforestation, land, tropical, areas	1.00, 0.55, 0.46, 0.42, 0.41
fastText	forest, deforestation, land, *land use*, tropical	1.00, 0.55, 0.46, 0.46, 0.42
fastText PatternRank 1	forest, deforestation, *land cover*, land, *land use*	1.00, 0.55, 0.50, 0.46, 0.46
fastText PatternRank 2 (Forest corpus)	forest, deforestation, *land cover*, land, *land use*	1.00, 0.55, 0.50, 0.46, 0.46
100 candidates		
fastText Keywords	forest, deforestation, land, degradation, tropical	1.00, 0.55, 0.46, 0.44, 0.42
fastText	forest, forest cover, *forest degradation*, rainforest, *tropical forests*	1.00, 0.95, 0.79, 0.76, 0.67
fastText PatternRank 1	forest, *forest cover*, *forest code*, *forest degradation*, *forest loss*	1.00, 0.95, 0.83, 0.79, 0.77
fastText PatternRank 2 (Forest corpus)	forest, *forest area*, *forest cover*, forestry, *forest management*	1.00, 0.98, 0.95, 0.89, 0.87

All the fastText models shown predict the correct top 1 annotation. However, they are more likely than the BERT models shown in Table 5 to also predict 'deforestation' as the second annotation in particular with 25 candidates. Using keyphrases and provideing 100 candidates alleviate this problem

candidates makes it more likely to have more similar annotations. Filtering out keyphrases based on their meaning remains a challenge.

We do not apply a threshold for the cosine similarity in our approach. By choosing a specialized corpus that is closely related to our images, and by choosing the candidates with keyword extraction, we ensure that the true labels with always be present in the candidate annotations. Furthermore, this specialized corpus makes it more likely to have candidate annotations that are highly relevant to the images beyond their labels. As can be seen in the examples shown in Tables 3, 4, 5 and 6, the cosine similarities of retrieved annotations is seldom below 0.50 for the top 5 annotations. When it happends, it seems to occur more specifically with keywords and ngram-based keyphrases and with only 25 candidates. Using a grammar rule to pre-select semantically valid candidates along with having a high number of candidates (100 or more) help keep annotation similarities at values above average. When we consider the list of candidates, we find that in many cases, when using keyphrases, we can get more semantic information compared to using keywords.

5 Conclusion

In this paper we propose a method to automatically annotate pairs of satellite images of forests with relevant keyphrases extracted from a specialized corpus. Our method combines visual semantic embeddings, keyphrase extraction and information retrieval. We evaluate our method on a dataset of image pairs from the Amazon rainforest and corpora of scientific publications related to deforestation. We show that keyphrases can provide rich semantic information without any negative impact on the annotation compared to annotating with keywords. We also show that fastText models outperform BERT models in terms of recall especially when the number of candidate annotations is high. One limitation of our work is that the keyphrase extraction methods do not significantly influence the annotation performance. Future work could investigate improvement to the visual semantic learning including the pre-training of networks on change detection. Another area for future work is the use of other keyphrase extraction methods along with the filtering of candidate annotations based on their meaning. The method presented here is applied to the specific case of deforestation, however it can be used in other domains where there is a need to add semantic information to image pairs such as the medical domain.

Funding Statement Open access funding provided by Université Toulouse III - Paul Sabatier. This project is funded in part by the AI4AGRI project (Horizon Europe, Grant ID: 101079136).

Data Availability Statement The Landsat images are available for download at https://earthexplorer.usgs.gov/. The labels are available from the data producers [10]. The scientific publications are available for download from the Web of Science at https://www.webofscience.com/.

Declarations

Conflict of Interest Statement The authors declare no conflict of interest.

References

1. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, *et al.* (2021) Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp 8748–8763 . PMLR
2. Neptune N, Mothe J (2021) Automatic annotation of change detection images. Sensors. 21(4):1110
3. Bouyerbou H, Bechkoum K, Benblidia N, Lepage R (2014) Ontology-based semantic classification of satellite images: Case of major disasters. In: 2014 IEEE Geoscience and Remote Sensing Symposium, pp 2347–2350 . IEEE
4. Bouyerbou H, Bechkoum K, Lepage R (2019) Geographic ontology for major disasters: methodology and implementation. Int J Disaster Risk Reduction 34:232–242
5. Uzkent B, Sheehan E, Meng C, Tang Z, Burke M, Lobell D, Ermon S (2019) Learning to interpret satellite images in global scale using wikipedia. arXiv:1905.02506.

6. Lu X, Wang B, Zheng X, Li X (2017) Exploring models and data for remote sensing image caption generation. IEEE Trans Geosci Remote Sens 56(4):2183–2195
7. Daudt RC, Le Saux B, Boulch A (2018) Fully convolutional siamese networks for change detection. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp 4063–4067 . IEEE
8. Daudt RC, Le Saux B, Boulch A, Gousseau Y (2019) Multitask learning for large-scale semantic change detection. Comp Vision Image Understand 187
9. Peng D, Zhang Y, Guan H (2019) End-to-end change detection for high resolution satellite images using improved unet++. Remote Sensing 11(11):1382
10. Bem PP, Carvalho Junior OA, Fontes Guimarães R, Trancoso Gomes RA (2020) Change detection of deforestation in the brazilian amazon using landsat data and convolutional neural networks. Remote Sensing 12(6):901
11. Neptune N (2022) Bibliometrics, change detection and multimodal learning on earth observation data: The case of deforestation. PhD thesis, Université Paul Sabatier-Toulouse III
12. Neptune N, Mothe J (2023) Annotating satellite images of forests with keywords from a specialized corpus in the context of change detection. In: Proceedings of the 20th International Conference on Content-based Multimedia Indexing, pp 14–20
13. Schopf T, Klimek S, Matthes F (2022) Patternrank: Leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. In: Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - KDIR, pp 243–248 https://doi.org/10.5220/0011546600003335 . INSTICC
14. Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 3982–3992
15. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Associate Comput Linguistics 5:135–146
16. Wan X, Xiao J (2008) Collabrank: towards a collaborative approach to single-document keyphrase extraction. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp 969–976
17. Campos R, Mangaravite V, Pasquali A, Jorge A, Nunes C, Jatowt A (2020) Yake! keyword extraction from single documents using multiple local features. Information Sciences 509:257–289
18. Grootendorst M (2020) Keybert: Minimal keyword extraction with bert. Zenodo
19. Hulth A (2003) Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp 216–223
20. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26
21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
22. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Ranzato M, Mikolov T (2013) Devise: A deep visual-semantic embedding model. In: Advances in Neural Information Processing Systems, pp 2121–2129
23. Wang L, Li Y, Huang J, Lazebnik S (2018) Learning two-branch neural networks for image-text matching tasks. IEEE Trans Pattern Anal Machine Intell 41(2):394–407
24. Shimabukuro YE, Duarte V (2000) Kalil Mello. Presentation of the Methodology for Creating the Digital PRODES, EM, Moreira, JC
25. Akinyemi J, Mothe J, Neptune N (2018) Fouille de publications scientifiques pour une analyse bibliométrique de l'activité de recherche sur la déforestation. In: EGC-Atelier Fouille du Web, pp 11–23