University of Trento

Department of Information Engineering and Data Science

Phd Thesis

~ · ~

Academic Year 2024–2025

Advanced Methods for Remote Sensing Image Captioning

Supervisor
Prof. Farid Melgani

PhD Student
Riccardo Ricci

A mamma, papà e Benedetta.

Acknowledgments

When I sit back and reflect on my PhD journey, realizing that all of it happened within three years is difficult. How did I compress so many things in just a short time? Over these three years, I accomplished many things that, before starting my PhD, I never thought I would be capable of achieving. For this, I'm grateful to my professor and guide, Farid Melgani, who has always found the time to comfort me when I was down, cheer me when I was good, and guide me when I was lost.

I also thank the many people I encountered in my journey, and directly or indirectly left some memories in me that I will carry for the rest of my life.

I thank Genc, Fabio, and Praveen for the laugh and work together. I wish you all the best in your future careers and hope your dreams come true.

I thank Federico for the many lunches we shared together and for the laugh we had, also during difficult times.

I thank Devis Tuia for accepting my visiting period in his lab in Switzerland; it was an experience that I will forever remember.

I thank Valerie and Li to have shared their lab with me during my period abroad. Your kindness and attitude made me feel at home.

And last but not least, I want to thank my mother, Antonella, and my sister, Benedetta, for always being there for me. I want you to know I will always be here for you, too.

To my father Flavio, I would like to tell you that I often think about you and still have you here in my memory.

Abstract

Someone once said that "an image is worth a thousand words." This captures well the amount of semantic information hidden inside these matrices of pixels. Looking at an image instinctively induces us to form hypotheses about which objects are inside, their state, dislocation, etc. We thus create, in our head, a high-level semantic representation of the image, understanding its contents. This ability, although instinctive and almost effortless for us, is very difficult to reproduce inside machines. Researchers have worked to replicate human image perception for decades, initially focusing on categorizing images or detecting specific objects, resulting in semantic representations tied to fixed concepts. Image captioning (IC) involves generating natural language descriptions for images, enabling machines to communicate their perception through language. This approach provides a flexible framework to convey diverse semantics. Despite significant advancements, IC systems face challenges in flexibility and reliability, particularly in specialized domains like remote sensing (RS). The contributions presented in this thesis are organized into chapters, each addressing a limitation in remote-sensing image captioning (RSIC). Chapter 2 introduces the fundamentals of generative image captioning, while Chapter 3 explores two distinct approaches to enhance robustness and accuracy. First, we propose an ensemble method that leverages collective knowledge from multiple participants to improve the reliability of image captioning outputs. Second, we fine-tune a pre-trained large vision-language model using an instruction-based multi-task dataset, showing how pre-training on a large visual-language dataset results in a better adaptation to downstream tasks with limited labeled data. We further evaluate how integrating multiple tasks into the same framework influences single-task performance. Chapter 4 focuses on enhancing the richness and detail of image descriptions, addressing the limitation of current RS image captioning datasets, where complex scenes are often reduced to a single, simple sentence. We propose to simulate a visual dialogue between two pre-trained instruction following models to iteratively dig for more detailed information. Based on several metrics, we show that our paradigm can generate descriptions that can discriminate better between different scenes. Chapter 5 focuses on Visual Question Generation (VQG) for remote-sensing images, which aims at generating natural language questions for a given input image. We introduce a new dataset to overcome the lack of question diversity in existing RS-VQG datasets. We also train a vision-language model to generate questions directly from remote-sensing images. VQG can address a serious limitation of our visual dialogue paradigm, which generates questions from an initial image description. Directly generating the questions from the image reduces the risk associated with an incoherent first description of the image. Chapter 6 focuses on describing changes between pairs of remote-sensing images. We use entirely pre-trained models, eliminating the need for custom model training. We explore how the instructions provided to these models can steer the description toward particular aspects of interest for the user. Chapter 7 introduces our initial exploration of incorporating supplementary geographic information into a remote-sensing image captioning pipeline. We aim to generate more detailed captions tailored to the specific scene and its geographic features. We believe integrating GIS data into vision-language models can enhance their groundedness when solving different tasks. In light of recent advancements in large vision-language models, we think that our endeavors in image captioning reflect a gradual shift from task-specific models trained to perform single tasks to multi-task large vision-language models that can accomplish several tasks at the same time, framing each as an answer to a different user *instruction*. We think adapting at inference time to multiple tasks and requests, as we show in Chapters 3 and 6, represents a fundamental ability that future vision-language models should possess and can greatly benefit remote-sensing applications. Finally, we think that RS vision-language research should move toward the inclusion of additional data sources (i.e., geographic databases) to help vision-language models be more precise and grounded in answering specific image-related queries.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An image captures a snapshot of a scene at a certain moment, impressing it on a grid of millions of pixels. The numerical value of each pixel depends on how light interacts with objects and thus is scattered back to a photosensitive sensor, like the one we have on our cameras. Thus, pixel values shift as the scene changes, influenced by several factors, such as the objects, their state, conditions, arrangement, quantity, etc. Interpreting an image means analyzing a certain configuration of pixels to assign it a semantic meaning. Humans rapidly develop this ability: from an early age, we can recognize visual elements, their relationships, their alignment with prior knowledge, and various implicit contextual elements.

In the last decades, the sheer volume and speed at which images are generated have steadily increased, with an estimate of 61400 images every second in 2024 [1], and accounting only for those taken by smartphones or cameras. Additionally, vast amounts of image data are continuously generated from space exploration instruments, medical imaging technologies, surveillance systems, and other domains. At this rate, human interpretation alone is insufficient to ensure timely and consistent analysis for the numerous applications that use such data. Consequently, enabling machines with the capability to automatically interpret and understand images has become an important endeavor, creating a thriving research field. Early efforts, as illustrated in Figure 1.1, focused on machines to automatically categorize images into predefined classes. For example, studies such as [1] and [2] addressed binary classification problems, distinguishing cityscape-landscape or indoor-outdoor scenes. Subsequent research expanded this approach to several classes [3], eventually progressing to modern systems capable of categorizing images into hundreds or even thousands of classes [4]. However, often, categorizing images into mutually exclusive classes is restrictive. To increase interpretability, researchers have thus developed multi-label classification algorithms [5][6], which assign a set of representative labels, allowing images to belong to multiple categories simultaneously.

As the request for finer image interpretation techniques advanced, object detection was developed to identify and highlight individual objects within images. Early methods were primarily focused on detecting faces [7][8], with modern approaches that can now recognize and detect thousands of different objects, accommodating variations in scale, pose, and appearance [9]. Image segmentation has also been proposed as a logical extension of object detection, offering a more precise approach by delineating objects through full segmentation rather than relying on bounding boxes or other coordinate-based representations [10]. This technique can also segment different semantic regions within an image, such as distinguishing between buildings, roads, vehicles, trees, and other elements [11].

While the aforementioned techniques prove valuable tools for image interpretation, their outputs are not always accessible to human observers. This is especially the case for specialized areas like the medical field, where automatic image interpretation outputs are rarely auto-explicative of their meaning. Another

[1] "Photo Statistics: How Many Photos are Taken Every Day?", Matic Broz, Photutorial.

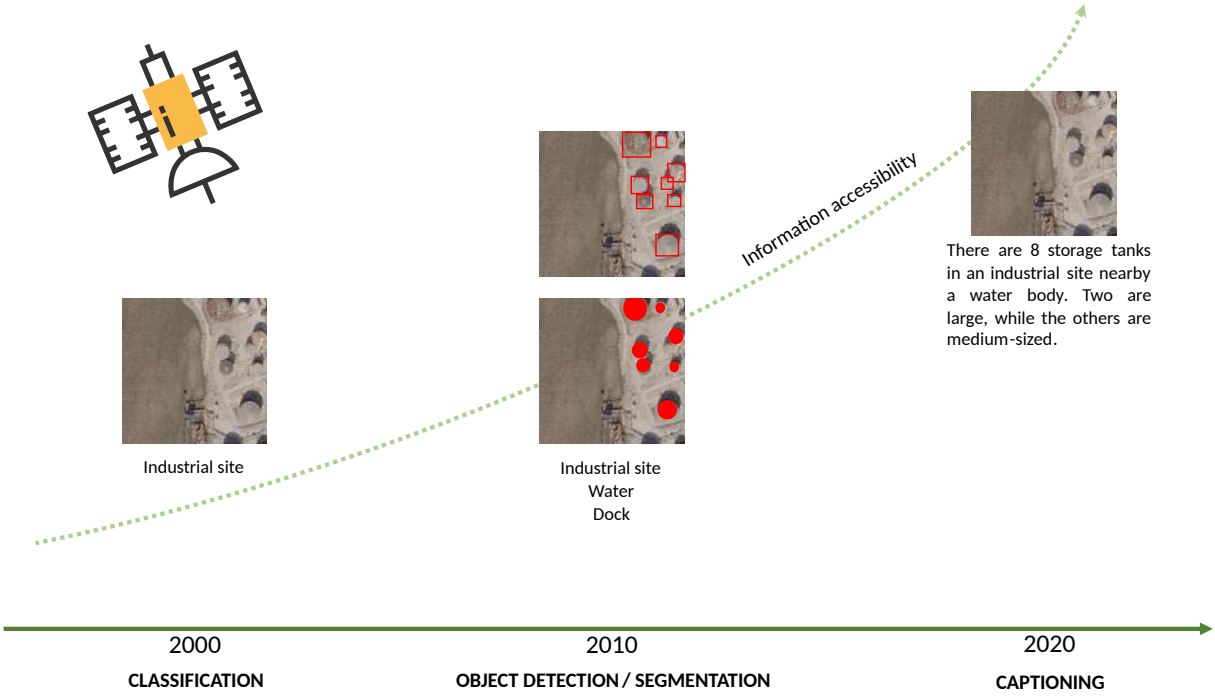


Figure 1.1: The evolution of image interpretation in the last decades. Both granularity and accessibility to the meaning of the extracted semantics increased over time.

example is remote sensing images, which can capture light in wavelengths different from what we are used to and thus look incomprehensible to non-experts.

Due to his flexibility, humans rely on *language* as the primary mode of communication. Language allows for the direct conveyance of complex semantic information, often bypassing the interpretative challenges associated with other forms of expression by presenting content in a readily comprehensible format. For this reason, the field of *image captioning* has emerged, aiming to develop machines capable of extracting semantic information from images in natural language descriptions. In the example of image description, natural language can convey detailed information, such as objects' proximity, relative positions, sizes, quantities, and other contextual details that are challenging to capture through other types of outputs. Image captioning can thus be viewed as the merging of several different tasks together: object recognition and counting, object attribute detection, and relative position understanding.

Beyond providing a more direct and accessible interpretation of image contents, captioning can be useful in several applications, such as text-to-image retrieval [12][13], precision agriculture [14][15], human-machine interaction [16][17], traffic data analysis [18], medical applications [19][20], helping visually impaired people [21][22][23], improving dataset quality [24] and others.

The first approaches to image captioning were based on templates [25][26][27], in which a machine detects concepts and phrases from the image and then composes them to generate a description. Although simple, this method tended to create repetitive captions, lacking the natural feeling of a human description. Another set of techniques is based on retrieval, framing image captioning as a ranking problem [28]. The general idea is to have a dataset of images and their respective captions [29], and with the target image, search for the most similar image within the database and transfer the caption from the database to the output. Most of the architectures in this category embed images and texts in a latent space, where the search is conducted [30][31]. Although this can enhance the naturalness of the generated caption,

it still has drawbacks when the algorithm is presented with images that differ significantly from images in the database. In that case, the algorithm is prone to failure, as it can assign a caption that is not necessarily coherent with the target image.

A third category, generative image captioning, has emerged to address the limitations of template-based and retrieval-based approaches. This method has been extensively researched because it can mimic human writing style, generate fluent text, and create novel captions by combining concepts learned during training. Generative image captioning relies on *auto-regressive* language models, which generate text by sequentially predicting the next word in a sentence given the context of the previous words and the image. This approach imposes no pre-defined constraints on text generation, giving freedom to the model to generate text in an open-ended manner. The greater flexibility of this framework has led to its widespread adoption in nearly all recent image captioning applications. These applications commonly use neural auto-regressive language models, which approximate the next-word distribution using neural networks.

Early approaches to generative image captioning [32][33] framed the task using a two-step framework. The process involves an *encoding* stage, where the image is analyzed to produce a condensed semantic representation, followed by a *decoding* stage, in which text is generated based on the extracted visual semantics until a stopping criterion is met, indicating the completion of the caption. When these pioneering works were conducted, convolutional Neural Networks (CNNs) were the default choice for handling visual inputs. Global features were extracted by feeding the image into the CNN and keeping the output from the layer immediately preceding the classification step. For example, in [33], the authors use global features from GoogleNet as the initial hidden state for an RNN-based language model. In [32], the authors use global features from VGG [34], injecting them at each step of the language model. The focus of this thesis is on remote-sensing (RS) image captioning (IC), specifically on the task of describing the content of RGB images as those depicted in Figure 1.2. These images are captured using sensors similar to standard cameras or smartphones. What distinguishes them is the acquisition angle and resolution: they are taken from an overhead perspective using aircraft, drones, or satellites, and they usually cover vast areas on Earth.

1.1 Evolution of RS image captioning

First works in remote sensing image captioning [35][36] proposed pioneering datasets for the task, adopting an encoder-decoder configuration using global features from a CNN. Other works in this direction try to leverage global features but with a focus on a better alignment using either auxiliary modules like a variational auto-encoder (VaE) in [37] or by leveraging pre-trained models like CLIP [38] which are trained in a contrastive learning setting to align images and texts in a shared latent space. Given the peculiarity of remote sensing images, multi-scale approaches have been explored. The idea is to leverage intermediate layers of a CNN to capture multi-scale information. The hierarchical structure of CNNs preserves the spatial coherence between regions in the input image, and the deeper the layer, the more the features are aggregated in a coarser scale. An example is [39], in which the authors use a multi-scale cropping mechanism that extracts features at three different scales before resizing them to obtain a fused feature representation covering different scales. Following the introduction of the attention mechanism, many researchers adopted it in the context of RS image captioning. For example, in [40], authors use a multi-level attention mechanism to focus on different parts of the images and words of the generated sentence. These two levels of attention are combined using another attentive layer to yield the final features used to condition a long-short-term memory (LSTM) decoder that generates the caption. In [41], the attention mechanism adaptively aggregates image features of specific spatial regions and scales. The authors introduce a contextual attention module to align visual scenes and attribute information. In [42], the authors use a cross-hierarchy attention mechanism, in which feature representation of objects

Figure 1.2: Examples of RGB remote sensing images of varying resolution. The first row contains low-resolution images that thus cover a larger area. The second row contains high-resolution images, where single cars can be spotted, covering more contained areas.

and patches are extracted using the region proposal of an object detection algorithm. These are further complemented by the global feature vector, and an attention module is used to fuse the information from different hierarchical regions of the image. After the advent of the Transformer [43], several approaches switched to using this architecture for remote sensing image captioning. In [44], the authors propose the CapFormer, a purely transformer-based image captioning approach for remote sensing. In [45], the authors propose a transformer-based architecture in which the decoder can adaptively focus on multi-scale features fused using topic-sensitive word embeddings. To better separate foreground from background in images, in [46], the authors use a transformer-based approach equipped with deformable scaled dot-product attention to mine multi-scale features from foreground and background areas. To better fuse visual and linguistic features during caption generation, authors of [47] propose a cross-modal reasoning dual transformer model. Specifically, they use a Swin Transformer (SwinT) encoder to better model multi-scale visual features and discover the intrinsic relationship in the objects. A transformer-based decoder is equipped with a cross-attention component to attend to visual features coming from the encoder. Besides architectural improvements, some works focused on using additional information inside the RS image captioning pipeline. In [48], the authors explore using annotated image regions to better direct the model's attention to specific regions of interest during caption generation. They use region-grid

features and geometry relationships to estimate correlations between different regions in the image. In [49], the authors use additional information from the predicted image label word embedding to guide the calculation of attention masks. In [50], a small target attention module calculates the attention weights using additional word embeddings of names of small targets that are difficult for the model to include in the output caption. Differently from integrating additional information, some works explore using multi-task settings, integrating auxiliary tasks to aid remote sensing image caption generation. Similarly, authors of [51] and [52] explore multi-label classification as an auxiliary task. The authors use multi-label scene classification due to its conceptual similarity to image captioning and its capability of highlighting semantic classes. In [53], a meta-learning strategy is tested, using natural and remote sensing image classification as meta-tasks. The idea is to leverage meta-features from classification models as input to the image captioning model. Using meta features already adapted to remote sensing image classification, the model requires a relatively small amount of caption-labeled data to achieve effective performance on RSIC. In [54], visual question answering (VQA) is explored as an auxiliary task for remote sensing image captioning. According to the authors, VQA aids the model in better-differentiating images, which is a limitation in RS due to the high inter-class similarity. With the advent of the Transformer and the discovery of auto-regressive language modeling as a strong unsupervised signal for implicit multi-task learning, several large language models have been built and trained on massive amounts of data. Due to their compatibility with solving vision language tasks, large language models have been adopted for remote sensing image captioning. Early works, such as [55], leverage pre-trained large language models (LLMs) to generate captions for remote sensing images by describing their object annotations. Due to their large-scale pre-training, large vision language models, and the expansion of LLMs to accommodate visual inputs, they have shown impressive capabilities in zero-shot settings and when finetuned on relatively small amounts of data. In [56], the authors harness the power of pre-trained large language models and powerful image encoders pre-trained using contrastive learning to achieve impressive image captioning and retrieval performance. However, due to the resource constraint in remote sensing tasks, other attempts tried to fine-tune a vision-language model already pre-trained to solve vision-language tasks in the natural domain. An example is [57], in which the authors curated a large-scale dataset of image-instruction pairs and fine-tuned a pre-trained LLaVA model on various tasks in a unified framework, achieving impressive capabilities. This approach is similar to [P1] though extended with further vision-language remote sensing tasks. Due to the advantage compared to training from scratch, other approaches followed the same paradigm [58][59][60], fine-tuning a pre-trained large vision-language model on instruction-based multi-task datasets. The large-scale language modeling pre-training injects a general world knowledge inside the language model that, by learning to predict the likely continuation of a text, learns to link textual concepts and thus their underlying semantics. Starting from the pre-trained model, fine-tuning on specific instructions becomes more effective and necessitates a relatively small amount of examples to achieve excellent performance. Furthermore, framing several tasks into the same format is beneficial as more data can be gathered and used to increase the generalization and accuracy of models on a varied suite of remote sensing tasks. Furthermore, these systems are highly efficient as the same model can solve different tasks within a unified framework.

Despite great advances in remote sensing image captioning and, more generally, in remote sensing visual language tasks, we think some critical aspects need further analysis. Despite the advances achievable using large pre-trained vision-language models, remote sensing image captioning remains an area with a scarcity of labeled data. Due to the cost of manual labeling, the datasets proposed for the task are relatively small, even though progress is being made in this direction. Thus, we think that a major concern in RS image captioning is robustness, as real-world applications necessitate accurate and consistent models. Beyond a pure data perspective, we think that to increase the groundedness of vision-language models in RS applications, integrating geographic information systems (GIS) is an interesting avenue. A vision-language model seamlessly interacting with a GIS platform represents a valuable advancement, especially for RS applications. Data-wise, another limitation of current RS image captioning lies in the

format of existing image captions. Most datasets contain short captions, often limited to a single sentence describing the primary feature within the image. Besides conveying more information to the end user, more detailed descriptions would prove favorable for several applications, such as text-to-image retrieval and generation, as they constitute a more complete semantic representation of the image contents. However, manually labeling a dataset of rich captions is a time-consuming task. We think leveraging current vision-language models' pre-trained capabilities is a promising direction to enrich remote-sensing image captions without creating apposite datasets. With the widespread adoption of multi-task large vision-language models, evaluating their output for tasks like image captioning remains challenging. Researchers discovered that standard metrics frequently evaluate a description based not on its intrinsic quality but on how closely it matches the language patterns of the reference dataset. This stems from the heavy reliance on exact word matching, which fails to account for variations in phrasing, such as long versus concise captions or different sentence structures. We think that addressing this limitation could represent a huge advancement in understanding how well a model describes the contents of an image. We think new evaluation metrics should prioritize semantic meaning over exact word matching. Finally, despite being highly task-efficient, large vision-language models are resource-intensive, limiting their large-scale applicability. We think research on efficient smaller vision-language models would be important, especially for adopting them on resource-constrained hardware settings such as drones or satellites.

1.2 Thesis objectives

This thesis is structured into self-contained chapters; each presenting research focused on analyzing a limitation of the current remote sensing image captioning field. The second chapter briefly overviews the fundamentals of generative image captioning and introduces important concepts that will be used throughout the thesis. The chapter begins by framing generative image captioning as a specialized form of language modeling, tracing the influence of advancements in natural language processing (NLP) that have reshaped the field and transformed its paradigm in recent years. Additionally, it presents the concept of language as a unified representational format capable of integrating diverse tasks within a single framework—a foundational principle driving the development of large vision-language models as versatile assistants for various applications.

Building on the robustness concerns expressed in the introduction, the third chapter focuses on methodologies that increase the accuracy and reliability of remote sensing image captioning. The hallucination problem, when a vision-language model generates text that doesn't accurately reflect the image's content, poses challenges in adopting such systems in real-world applications. Furthermore, a hallucination is often difficult to spot, as those models generally create fluent and coherent text. To increase robustness, we first propose to leverage the ensemble concept, where not a single one but several models are asked to produce a caption for a target image. The generated captions are then analyzed to isolate the final best caption. Secondly, following the practice of fine-tuning pre-trained large vision language models, we explore the use of multi-task learning and large vision-language models (LVLM) to boost captioning performance.

Based on our observation of a degree of expression restrictiveness in the target captions in the most famous remote sensing captioning datasets, in the fourth chapter, we explore ways to enrich the final caption using multiple turns of a machine-to-machine dialogue about the image content, sparked by the initial caption provided by a large vision-language model.

The fifth chapter is devoted to another vision-language task, visual question generation (VQG). We explored this field after recognizing visual question generation as an important building block of our dialogue system. There, the question generation was carried out in a pure text manner, which caused challenges when the first description was not pertinent to the image content. Here, the idea is to have models able to generate questions directly from the image itself.

The sixth chapter explores change captioning, a task in which the algorithm is asked to process multi-temporal images and generate a text that describes what changed during the time frame. Based on the restrictiveness of current datasets for this task, we analyze the application of pre-trained large vision-language models in this context, including our machine-to-machine visual dialogue. We analyze the ability to target the extraction of specific aspects of interest for the user at inference time.

The final chapter presents initial explorations into integrating ancillary geographic information to enhance remote-sensing vision-language tasks. Specifically, we investigate enriching image captions with OpenStreetMap (OSM) data and introduce a manually curated dataset of image-caption-OSM triplets to support this task. Our analysis highlights the complexities of integrating diverse data modalities and identifies challenges stemming from the inclusion of additional, scene-dependent information. Despite these issues, we emphasize the potential benefits of incorporating grounded, detailed geographic context to improve the utility and reliability of remote-sensing vision-language models. Finally, we propose future research directions to address these challenges, proposing more effective approaches to leverage the supplementary data.

Finally, based on the progression of the field in the last years, we draw our conclusions and highlight possible future directions.

Chapter 2

Fundamentals of Generative Image Captioning

The low naturalness and variety of template-based approaches and the costly maintenance of a large and representative database for retrieval-based approaches led researchers to develop a third category of methods, the generative approaches, based on auto-regressive language models. Generative image captioning has been deeply investigated for its ability to mimic human writing style and fluency and the possibility of generating novel captions by merging concepts learned during training. As depicted in

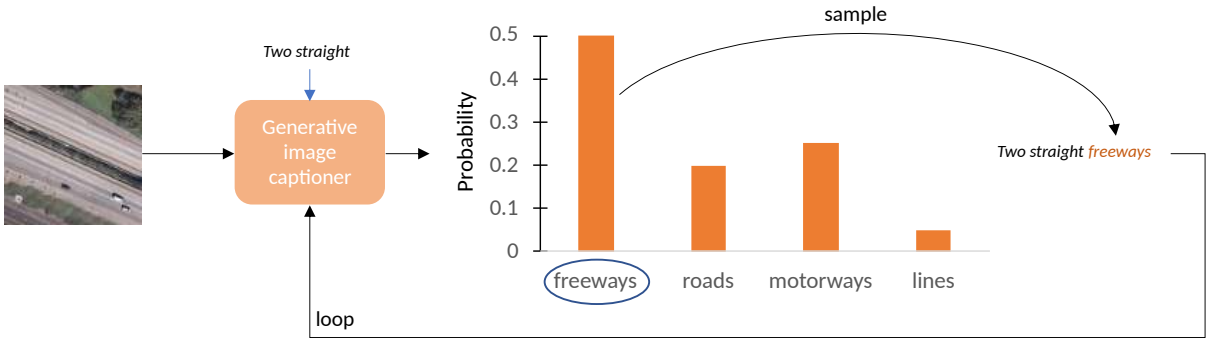


Figure 2.1: Visual representation of generative image captioning.

Figure 2.1, in generative image captioning, the prediction is carried out iteratively, segment by segment. The model assesses the probability for each possible next text segment at each iteration, leveraging contextual information from the previously generated text and the image. The model selects the most appropriate next text segment by sampling from this probability distribution, which is then appended to the generated sequence. This iterative process continues until a stopping criterion is met, signifying that generation is complete.

Thus, generative image captioning is an extension of *language modeling*, augmented with the integration of visual information derived from the image.

2.1 Tokens and tokenization

In auto-regressive language modeling, what we have thus far referred to as "segments" of text is termed *tokens*. A piece of text is thus converted into a sequence of tokens. Tokens represent the fundamental unit of information, and their value can vary depending on how we choose to split text. Some examples of splitting strategies are shown in Figure 2.2, and take the name of *tokenization* strategies. Each token is then mapped to a unique numerical identifier within a *vocabulary*. In that way, texts can be split into tokens and then converted to IDs, resulting in sequences of numbers that a neural network can process. This is a lossless conversion, as the same text can be recovered from the sequence of IDs with the inverse operation. As in Figure 2.2, two natural ways to split a text would be by individual words or

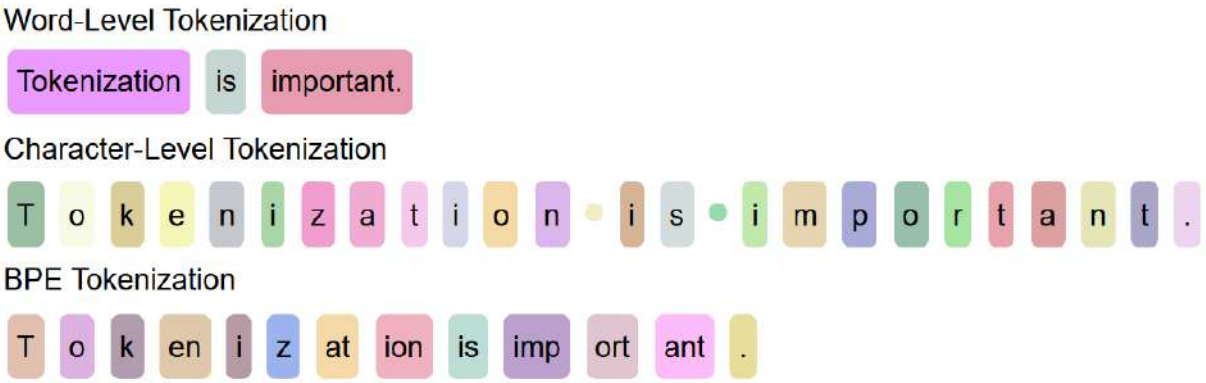


Figure 2.2: Example of different tokenization criteria on the same text. BPE is trained on text from the book "Harry Potter - The Sorcerer Stone."

individual characters. However, both approaches have limitations. In auto-regressive language modeling, it is crucial to balance the vocabulary size and the length of the sequence of IDs after text conversion. Word-level tokenization often leads to an excessively large vocabulary, which can dilute the information conveyed by each token. Conversely, character-level tokenization reduces vocabulary size but results in longer sequences of token IDs, which can lead to memory-related challenges.

More advanced tokenization methods have been developed to address these challenges by leveraging Byte Pair Encoding (BPE) [61]. BPE tokenizers decompose text into *sub-word* units—sequences of consecutive characters learned from a given text corpus and optimized for compressing text within a constrained vocabulary size. In the BPE algorithm, the most frequent byte sequences are iteratively merged into tokens, with this merging process continuing until the desired vocabulary size is achieved. For instance, the BPE tokenizer used in GPT-2 [62] employs a vocabulary of 50257 tokens. This vocabulary consists of 256 base byte tokens, a special end-of-text token, and additional tokens formed through 50000 merge operations.

2.2 Mathematical formulation

In auto-regressive language modeling, the likelihood of observing a text sequence $\mathbf{T}$, consisting of N tokens $\{t_1, t_2, \ldots, t_N\}$, is given by:

$$P(\mathbf{T}) = P(t_1, t_2, \ldots, t_N) = \prod_{i=1}^{N} P(t_i \mid t_{<i}) \tag{2.1}$$

The joint probability of observing the exact sequence of tokens in $\mathbf{T}$, $P(t_1, t_2, \ldots, t_N)$, is therefore expressed as the product of the conditional probabilities of each token t_i, given the preceding ones $t_{<i}$. A language model seeks to model the probability distribution over a large corpus of texts, denoted as $\{\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}, \ldots\}$, to maximize the likelihood of observing text samples drawn from this corpus. For instance, a language model trained on English text data is more likely to assign a higher probability to the sentence "a cat sitting on the sofa" than to "a cat sleeping in the fridge." This preference arises because the former is statistically more frequent to encounter compared to the latter.

In the paradigm of neural language modeling, which we adopt in this thesis, the probability distribution is modeled using a large neural network parameterized by θ. Given a dataset $\mathcal{D} = \{\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}, \ldots\}$ of texts, training a neural language model involves determining the parameters θ^* that maximize the following log-likelihood function:

$$\theta^* = \arg\max_{\theta} \sum_{\mathbf{T} \in \mathcal{D}} \log\big(P(\mathbf{T})\big) = \arg\max_{\theta} \sum_{\mathbf{T} \in \mathcal{D}} \sum_{i=1}^{N} \log\big(P(t_i \mid t_{<i}, \theta)\big) \tag{2.2}$$

in which the sum of the log probabilities replaces the product of probabilities without altering the problem's solution. Thus, the language model compresses the dataset $\mathcal{D}$ into (θ).

A trained neural language model can generate text by iterative sampling from its learned language distribution, as illustrated in Figure 2.1. Since the output of the language model represents a probability distribution over the vocabulary of possible tokens, various strategies can be employed to sample the next token. The simplest approach is to select the token with the highest probability at each step, a technique referred to as *greedy sampling*, depicted in Equation 2.3.

$$\hat{t_i} = \arg\max_{t_i} P(t_i \mid t_{<i}, \theta) \tag{2.3}$$

Other methods for text generation include beam search, nucleus sampling [63], top-k sampling, and others. These methods can be categorized into deterministic and non-deterministic approaches. Deterministic approaches, as the name implies, always produce the same output given the same input. Greedy sampling and beam search are examples in this category. On the other hand, non-deterministic approaches sample the next token based on the probability distribution generated by the language model, resulting in varying outputs for subsequent runs using the same input. Nucleus sampling and top-k sampling fall into this category. In the thesis, we focus exclusively on deterministic approaches to ensure the reproducibility of the results.

2.2.1 Language model (LM) vs. vision-language model (VLM)

What we have discussed so far applies to pure neural language modeling. However, the equations can be readily extended to incorporate the influence of an image $\mathbf{X}$. For instance, given an image $\mathbf{X}$ and a corresponding text $\mathbf{T}$ (such as its description), the probability of observing the description $\mathbf{T}$ subject to image $\mathbf{X}$ can be formulated as:

$$P(\mathbf{T} \mid \mathbf{X}) = P(t_1, t_2, \ldots, t_N \mid \mathbf{X}) = \prod_{i=1}^{N} P(t_i \mid t_{<i}, \mathbf{X}) \tag{2.4}$$

Similarly, given a dataset $\mathcal{D} = \{(\mathbf{X_1}, \mathbf{T_1}), (\mathbf{X_2}, \mathbf{T_2}), (\mathbf{X_3}\mathbf{T_3}), \ldots\}$ of images and paired texts, a neural vision-language model is defined by the parameters θ^* that maximize the following log-likelihood:

$$\theta^* = \arg\max_{\theta} \sum_{\mathbf{T}, \mathbf{X} \in \mathcal{D}} \log\big(P(\mathbf{T} \mid \mathbf{X})\big) = \arg\max_{\theta} \sum_{\mathbf{T}, \mathbf{X} \in \mathcal{D}} \sum_{i=1}^{N} \log\big(P(t_i \mid t_{<i}, \mathbf{X}, \theta)\big) \tag{2.5}$$

Sampling equations can be modified accordingly.

2.3 Modeling the next-token probability distribution

In neural vision language modeling, which includes but is not restricted to image captioning, the probability distribution is modeled using an encoder-decoder architecture. The encoder is a vision module responsible for compressing an image $\mathbf{X}$ into a manageable vector representation $Z_x \in R^{d_x}$, condensing the semantic information from the pixel level to a more abstract semantic level. Vector representations of data, like Z_x, are called *embeddings* and are characterized by their dimension, in this case d_k. The decoder is an auto-regressive language model that generates the caption token by token, conditioned on the image information (condensed in the embedding Z_x) and the preceding tokens.

2.3.1 Auto-regressive language model

One of the ubiquitous choices for the decoder, until 2017, was represented by recurrent neural networks (RNNs) [64]. RNNs are a class of neural networks designed to handle sequential data. These networks process input sequences one step at a time, maintaining a hidden state h_i that acts as the network's "memory" at step i. The network uses the hidden state to retain information about the previous steps $< i$ to condition the output at the current step i. The mathematical equation governing a plain RNN is the following:

$$h_i = f(W_{th}t_i + W_{hh}h_{i-1} + b_h + b_t) \tag{2.6}$$

Where W_{th} is a weight matrix controlling the information flow from input to the hidden state, W_{hh} is the weight matrix controlling information flow between hidden states at different steps, b_h and b_t are bias terms, and f is a non-linear activation function. Additionally, t_i represents the input at time step i.
The output at each time step i is calculated as:

$$y_t = g(W_{hy}h_t + b_y) \tag{2.7}$$

where W_{hy} is the hidden-to-output weight matrix, which projects the hidden state h_i to a probability distribution over the vocabulary, b_y is the bias term, and g is an activation function (e.g., softmax for classification).

Due to their recurrent computation, RNNs suffer from optimization issues. For the chain rule of derivation, the gradients at the earlier time steps become a multiplication of contributions from each following time step. This can result in gradients that either diminish to negligible values (vanishing gradients) or grow uncontrollably large (exploding gradients), making learning long-range dependencies harder. Two main variants have been proposed to counteract this issue: long-short-term memory (LSTM) [65] and gated recurrent unit (GRU) [66]. While their structural mechanisms share similarities, GRUs are often favored for their computational efficiency, offering performance comparable to LSTMs with reduced complexity.

The attention mechanism

In 2015, a paper [67] introduced a novel mechanism to address a critical limitation in Recurrent Neural Networks (RNNs). The authors drew inspiration from machine translation, where the standard approach involved an RNN encoder compressing a source sentence into a fixed-size hidden state h and an RNN decoder generating the translation conditioned on h.

The paper highlights a fundamental issue: the fixed size of h imposes a bottleneck on the architecture, as it requires the network to encapsulate information from the *whole* source sentence into a single, fixed-length representation. They showed how this approach often leads to performance degradation, particularly when handling longer sequences than those used to train the machine. If the network can adaptively absorb only the relevant context at each decoding step, it can address the bottleneck of storing all information in a single fixed representation, as different parts of the source sentence are likely to be

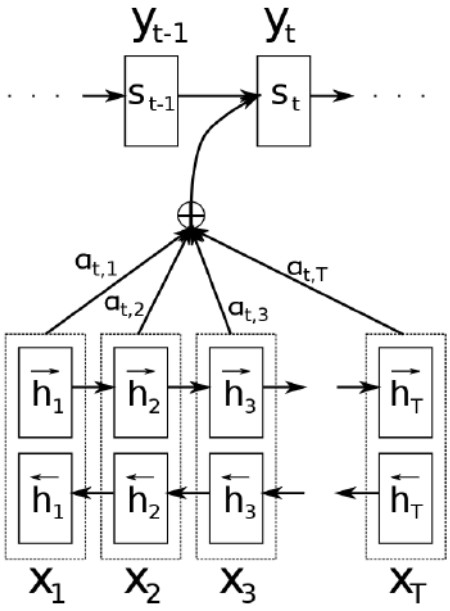


Figure 2.3: The "Bahdanau" soft-attention mechanism. When generating the target translation token y_t, the decoder adaptively integrates context from the source sentence by computing different "attention" scores for each hidden state $h_t, t \in [1, T]$ of the encoder.

relevant for different target token predictions. Thus, the authors introduced an "alignment" component that allows the network the possibility to generate a different context vector c_t at each decoding step via *soft alignment*. The authors applied the alignment component to an encoder-decoder RNN framework. As depicted in Figure 2.3, the prediction of the t-th target token y_t depends on the previous token y_{n-1}, the decoder hidden state s_t and a context vector c_t. In turn, the context vector c_t exerts its influence on the decoder hidden state s_t via

$$s_t = f(s_{t-1}, y_{t-1}, c_t) \tag{2.8}$$

The context vector c_t is adaptive and depends on the encoder hidden representations $(h_1, \ldots, h_T)$. The context is calculated as:

$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j \tag{2.9}$$

where each weight α_{tj} is normalized between 0 and 1 using a softmax function, which also ensures that $\sum_{j=1}^{T} \alpha_{tj} = 1$.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})} \tag{2.10}$$

where $e_{ij} = a(s_{i-1}, h_j)$. a is the alignment model, parametrized as a feed-forward neural network that is jointly trained with the other components of the system.

This marked the initial integration of an attention mechanism within language models—a development that would revolutionize the field within just a few years. Thanks to its natural property of allowing a model to adaptively "focus" on different parts of the input, thus mimicking human attention when looking at an image, researchers started to employ the attention mechanism for image captioning [68][69], reporting improved performance and a higher interpretation of the results thanks to the possibility to visualize the regions of the image "attended" by the network during caption generation.

The transformer architecture

Despite significant advancements in RNNs coupled with the attention mechanism, their sequential computation was still a source of limitations. In 2017, Vaswani et al. [43] introduced a radically different paradigm for language modeling. They introduced a novel architecture called *Transformer*, a sequence-to-sequence model that entirely eliminates recurrence.

A transformer architecture takes in input a sequence of N embeddings $Z^{in} = \{z_1^{in}, z_2^{in}, \ldots, z_N^{in}\}$ and outputs a sequence of embeddings of the same length $Z^{out} = \{z_1^{out}, z_2^{out}, \ldots, z_N^{out}\}$. The transformer comprises a stack of layers, each doing the same operation, depicted in Figure 2.4, called *dot product* attention. The output sequence has the same dimensions as the input sequence when fixing $d_{in} = d_{out}$, but its elements are a *transformed* version of the input sequence. Each output element results from aggregating information from all the elements in the previous layer. The aggregation is dynamically controlled using attention. Thus, it is data dependent, allowing the network to adaptively attend to the most relevant context for the particular example. The removal of the recurrence in transformers brings two advantages compared to RNNs. Firstly, transformers do not need to wait for previous steps to be completed to compute successive elements in the sequence but can perform the computation in parallel. This makes transformers more training efficient than RNNs. Secondly, each element in the output sequence is generated using the same number of operations, irrespective of its position. The number of operations depends only on the number of attention layers inside the network. As a result, each element in the input sequence potentially has the same possibility to influence the output, facilitating more effective modeling of long-range dependencies.

Transformers, however, have also drawbacks compared to RNNs. Given their formulation, the memory utilization scales quadratically with the input sequence length (think of the scores matrix, which is an $n \times n$ matrix). This is also true during inference, where RNNs are much quicker and memory efficient. Another drawback is that a transformer with plain attention layers cannot differentiate between sequence arrangements. For example, the output for a sequence $Z = \{z_1, z_2, \ldots, z_N\}$ will be identical to the output for a different arrangement of the sequence $Z = \{z_N, z_1, \ldots, z_2\}$, due to the linear combination of the elements. This is why, in transformers, there is the need to explicitly add position information inside the input sequence.

However, the higher parallelization reachable with transformers and the reported better performance on a wide range of tasks made this architecture the go-to choice for many researchers in the field as the standard network for language modeling.

2.3.2 The era of large language models

At the same time, language was starting to be used as an *universal* medium to seamlessly integrate different tasks into the same format. For example, in [70], authors used language to frame 10 different tasks, from translation to summarization to text comprehension. In 2019, another paper [62] demonstrated that auto-regressive language modeling of large quantities of web-scraped text results in unsupervised multi-task learning. The authors trained a transformer-based language model to perform next-token prediction on WebText, a dataset of millions of web pages. The model, without any explicit supervision, achieved state-of-the-art results on 8 out of 7 benchmarks in a zero-shot setting. The model was GPT-2, a $1.5B$ parameters transformer large language model, which marked the beginning of the era of large language models.

Foundational language models

This ability of large language models to gain multi-task abilities in an unsupervised manner from large quantities of web-scraped text marked a paradigm shift in the whole community, especially under the view of language modeling as a universal pre-training task to acquire broad knowledge about our world.

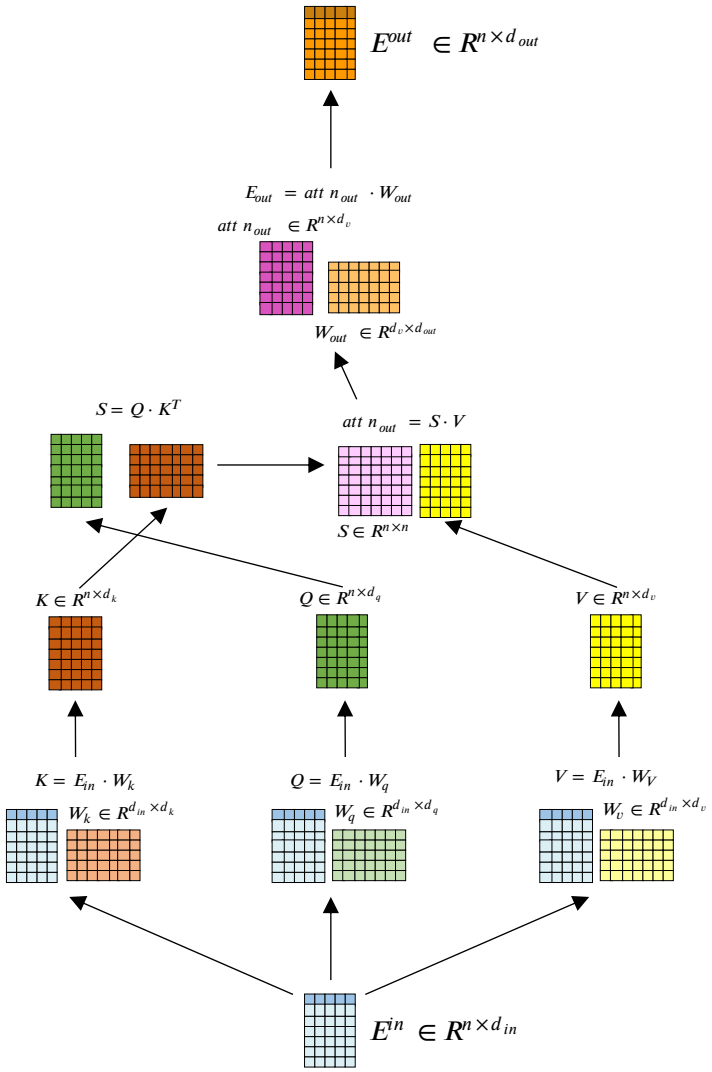


Figure 2.4: A single self-attention layer. The input sequence E^{in} is mapped to an output sequence E^{out}. Each element in the output sequence is a linear combination of elements in the value matrix (V) derived from the input sequence. The coefficients of the linear combination are determined by a dot product operation between the keys matrix (K) and the query matrix (Q), both derived from the input sequence. In cross-attention layers, the query matrix is derived from the input sequence, while the key and value matrices are derived from a different source (such as the output of an encoder).

GPT-2 was a very early demonstration of a *foundational model*. A foundational model is "any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks" [71]. Another paper [72] showed how training a bigger language model on more data greatly improves task-agnostic, few-shot performance, becoming competitive with fully supervised approaches on most datasets. Researchers found that some empirical formulas link language modeling performance to the model size, dataset size, and compute used during training [73]. Training a larger model on more data or for more iterations consistently leads to lower test set perplexity

(or greater language modeling performance). This, coupled with mostly exaggerated claims about AGI, led researchers and companies to invest in creating larger models and collecting larger datasets to train those models. Several orders of magnitude have spanned both model size and dataset size, creating foundational models showing impressive transfer capabilities to many downstream tasks.

Instruction fine-tuned language models

Through large-scale pre-training, foundational language models acquire a broad range of potential capabilities. However, these models necessitate fine-tuning using supervised data to correctly adapt to downstream tasks. The language modeling ability of the pre-trained foundational language model only implies that the model can generate a coherent continuation of a text sequence. This does not mean that they can follow the *intention* of a certain query. In the example in figure 2.5, the *prompt* is the query given to the language model. As humans, we can understand the request behind this prompt: the user wants an explanation of the moon landing suitable for a six-year-old kid. In the example completion, it is clear that the completion from the foundational language model (GPT3) does not *align* with our expectation. The model writes four sentences similar to the input but asks for other theories. This is an example in which the completion of the model does not "follow" the prompt formulated by the user, thus not carrying out the expected task. Differently, InstructGPT, an instruction-following model (an "aligned" model), correctly "understands" the request hidden in the instruction, providing an answer that effectively *solves* the task. The instructGPT model, nowadays known as ChatGPT [74] was released

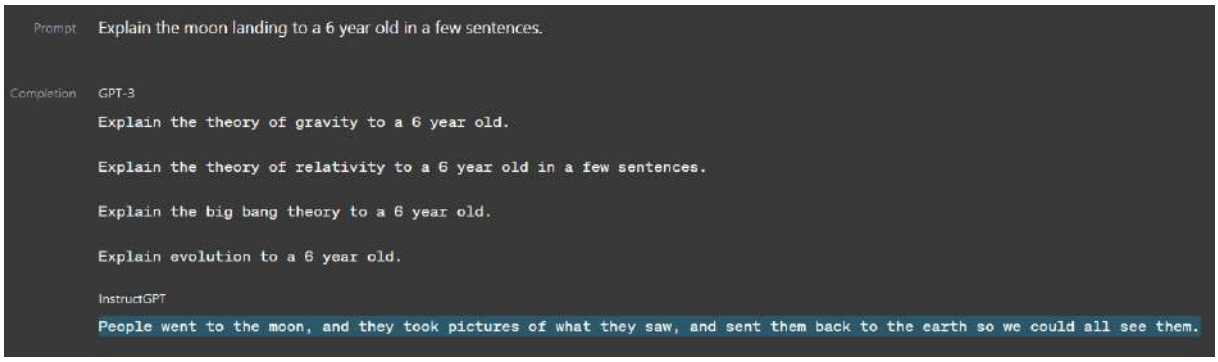


Figure 2.5: Example of different responses to the same query by a foundational model (GPT3 [72]) and an instruction fine-tuned model (InstructGPT [74])

in 2022, and was the first attempt to build an *instruction-following* model. If the model can correctly "follow" user intention, it is also an implicit multi-task solver within the same framework and, more importantly, without fine-tuning each downstream task. To align the model to follow instructions, the authors used a mix of supervised fine-tuning using conversational data labeled by humans and reinforcement learning from human feedback (RLHF), a technique to further push the model towards answers that are preferable to humans. An example of an instruction-response pair is provided below.

```
{
    "instruction": "Is the following statement true or false?
                    The human body has 207 bones.",
    "response": "false"
}
```

The potential of instruction-following models is the ability to solve various tasks at inference time without the need for supervised fine-tuning. The user can use the prompt to influence model behavior and how it

responds. For instance, using an instruction-following model, we can push the model to solve a sentiment analysis problem by prompting it with the following query: "Rate the sentiment of this statement: (statement) as positive, negative, or neutral. Respond with a single word."

Because these models are trained on extensive datasets containing human-like queries, they understand that the task is to evaluate the sentiment of the provided statement and respond with one of the specified options: "positive," "negative," or "neutral." In Chapter 3 and Chapter 5, we will see some applications of this capability to solve remote sensing tasks without the need to fine-tune the models on labeled data for the task. Of course, to better align with some tasks or situations, models can be fine-tuned using custom datasets, but the idea is that using language as the interface, a single model can be used to solve a plethora of requests without the need to collect any particular data for any specific task. These models have been commonly labeled as *assistants*. This discourse holds for vision-language models that can give rise to vision-language assistants when adapted to follow instructions involving visual content.

2.4 Metrics for image captioning

This section provides an overview of the commonly used metrics for image captioning. Mostly, research in image captioning adopted metrics based on exact word matching, though some also involve other strategies.

2.4.1 Bilingual evaluation understudy (BLEU)

BLEU [75] calculates a modified n-gram precision between a candidate text and a set of reference texts. The BLEU score is defined as:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.11}$$

Where BP represents the **brevity penalty**, which penalizes short translations in comparison to the reference. The variable w_n denotes the weights for the n-gram precisions, with the typical value of $w_n = \frac{1}{N}$, where N is the maximum n-gram length considered. The term p_n refers to the precision for n-grams of length n, which is the fraction of n-grams in the generated text that also appears in the reference text. BLEU is bounded between 0 and 1, with higher scores indicating a better match between reference and generated text.

2.4.2 Consensus-based Image Description Evaluation (CIDEr)

CIDEr [76] is also based on n-gram precision between the candidate text and a set of reference texts. Furthermore, it uses the TF-IDF measure to penalize n-grams that commonly occur across all references in the dataset. The CIDEr score referring to an n-gram is:

$$\text{CIDEr}_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\| \|g^n(s_{ij})\|} \tag{2.12}$$

Where c_i represents the candidate caption, and S_i is the set of m reference captions. The function $g^n(\cdot)$ denotes the TF-IDF weighted n-gram vector representation. $\|g^n(c_i)\|$ and $\|g^n(s_{ij})\|$ are the norms (magnitudes) of the n-gram vectors for the candidate caption c_i and the j-th reference caption s_{ij} in S_i, respectively. The dot product $g^n(c_i) \cdot g^n(s_{ij})$ measures the cosine similarity between the n-gram vectors. Finally, the term $\frac{1}{m}$ normalizes the score by the number of reference captions. The complete CIDEr

metric is an average of the score computed for n-grams from 1 to 4, as in the following equation

$$\text{CIDEr}(c_i, S_i) = \sum_{n=1}^{4} w_n \, \text{CIDEr}_n(c_i, S_i), \tag{2.13}$$

CIDEr is bounded between 0 and 5, with higher scores indicating a better match between reference and generated text.

2.4.3 METEOR

METEOR [77] is based on unigram matching between a candidate text and a set of references. Unigrams can be matched based on surface forms, stemmed forms, or meanings (using a synonym stemmer), and in the case of multiple references, an alignment module selects the best alignment between the generated text and each reference text. The alignment metric ensures that the words in the generated texts are arranged similarly to the reference text. The hypothesis is that similar arrangements correlate with a more syntactic and semantic agreement between the generated and the reference texts. After finding the best alignment, METEOR computes the harmonic mean between precision and recall.

$$F_{\text{mean}} = \frac{10PR}{R + 9P} \tag{2.14}$$

Where P is the precision computed as the ratio of unigrams in the generated text matched in the reference text over the total number of generated unigrams, and R is the recall computed as the ratio of unigrams in the reference text that is present in the generated text over the total number of unigrams in the reference text. METEOR gives higher weight to recall. METEOR further calculates a penalty score based on the alignment between the generated and the reference text, which evaluates worse alignments lower than better alignments. The final formula of METEOR is:

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - \text{Penalty}) \tag{2.15}$$

METEOR is bounded between 0 and 1, with higher scores indicating a better match between reference and generated text.

2.4.4 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE [78] is a set of metrics used to evaluate the coherence between a generated *summary* and a reference summary. We will specifically use ROUGE-L in the rest of the thesis. Generally, for each sentence r_i in a given reference text R composed of u sentences, the longest common sub-string (LCS) $LCS_{\cup}(r_i, C)$ match is calculated against the union of all sentences in the generated text C. The longest common substring is a series of adjacent words in the reference and the generated text. After the LCS has been calculated for each sentence in the reference text, precision and recall are calculated like:

$$R_{lcs} = \frac{\sum_i^u \text{LCS}_{\cup}(r_i, C)}{m} \tag{2.16}$$

$$P_{lcs} = \frac{\sum_i^u \text{LCS}_{\cup}(r_i, C)}{n} \tag{2.17}$$

Where m is the total number of unigrams in the reference text and n is the total number of unigrams in the generated text. Then, the ROUGE-L metric is computed as:

$$\text{ROUGE-L} = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{2.18}$$

Where $\beta = 1.2$ following the implementation of Salaniz[1] on GitHub. ROUGE-L is bounded between 0 and 1, with higher scores indicating a better match between reference and generated text.

[1] https://github.com/salaniz/pycocoevalcap

Chapter 3

Increase Captioning Accuracy and Robustness

[P2] Riccardo Ricci, Farid Melgani, Josè Marcato Junior, and Wesley Nunes Goncalves. "Robust Image Captioning with Post-Generation Ensemble Method". In: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium. IEEE. 2023, pp. 5234–5237.

[P3] Riccardo Ricci, Farid Melgani, Josè Marcato Junior, and Wesley Nunes Goncalves. "NLP-Based Fusion Approach to Robust Image Captioning". In: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2024).

[P1] Yakoub Bazi, Laila Bashmal, Mohamad Mahmoud Al Rahhal, Riccardo Ricci, and Farid Melgani. "Rs-llava: A large vision-language model for joint captioning and question answering in remote sensing imagery". In: Remote Sensing 16.9 (2024), p. 1477 (my contribution: idea conceptualization).

Introduction

This chapter is dedicated to methods that can increase the robustness and accuracy of remote-sensing image captioning. A significant problem in this field is the scarcity of large-scale human-curated datasets and the intrinsic complexity and variability within remote sensing scenes. Due to the cost, both monetary and in terms of time, of manually labeling RS scenes with descriptive captions, the available datasets are significantly restricted when compared to natural image captioning datasets [79][80], as can be seen in Table 3.1. To date, four datasets have been publicly released for remote sensing image captioning: Sidney-Captions [35], UCM-Captions [35], RSICD [36] and NWPU Captions [41]. The main reason for this difference is that it is easier to web-scrape samples of natural images and their descriptions than remote-sensing scenes. The confidence in building real-world applications is primarily linked to the reliability of a model. Still, we know that a scarcity of training data can harm both robustness and generalization. As we saw in the thesis introduction, most of the literature has focused on improving captioning performance by introducing several variations to the CNN-RNN vanilla pipeline. In this chapter, we propose and analyze two ways to enhance the robustness and performance of remote-sensing image captioning algorithms: 1. applying an ensemble of captioners and 2. training a large vision language model (LVLM) using instruction tuning on multiple tasks at once.

Table 3.1: Statistics of image captioning datasets in the computer vision (CV) and the remote sensing (RS) community.

	Name	N° of images	Captions per image
CV Datasets	COCO Captions [79]	328000	≈ 5
	Flickr30k [80]	31000	5
RS Datasets	NWPU Captions [41]	31500	5
	RSICD Captions [36]	10921	5
	UCM Captions [35]	2100	5
	Sidney Captions [35]	613	5

3.1 Ensemble for image captioning

Ensembles have been extensively studied in the literature to increase the accuracy and robustness of algorithms on classification problems [81]. For example, authors in [82] adopted an ensemble to boost the accuracy of change detection maps. In [83], the authors apply ensembles to increase multi-class classification accuracy. Despite the success of ensembles for classification, they received little to no attention in the context of image captioning. In [84], the authors apply an ensemble by summing the next-token probabilities of each captioner and sampling upon the aggregated probability distribution. As the authors showed, this approach only slightly boosts captioning performance; however, we argue that formulating an ensemble for image captioning in such a way poses significant constraints. First, all captioning models must use the same vocabulary. Employing differing vocabularies leads each captioner to generate probability distributions over different sets of possible tokens, making it impossible to sum individual probabilities before sampling the next token. This is a limitation since it is common for different pre-trained algorithms to employ different vocabularies. Second, most captioners must return coherent predictions; otherwise, there is a high risk of severely impacting the performance since every contribution is weighted equally. For these reasons, we propose a different ensemble paradigm, in which the idea is to leverage the ensemble a-posteriori. This means that ensemble techniques are applied after each model has generated its final description. With this formulation, we remove all the limitations: what is necessary is a generated caption of each model, irrespective of how it is obtained. In this way, there are no predefined constraints, leaving the freedom to (a) adopt different architectures, (b) adopt different caption generation schemes, and (c) use different vocabularies. We propose three methods, each with strengths and drawbacks, providing a thorough analysis of the a-posteriori ensemble in different scenarios. The main contributions can be summarized as follows:

- We propose three strategies to implement an ensemble in the context of image captioning.
- We evaluate our approaches using four remote sensing (RS) image captioning datasets, assessing the ensemble's robustness and performance across different scenarios.

3.1.1 Methodology

Our ensemble for image captioning is depicted in Figure 3.1. Our approach is divided into two main stages: caption generation and post-generation ensemble. During caption generation, a set of *different* algorithms is employed, producing a set of caption candidates, one for each participant. The post-generation ensemble stage ingests the set of candidates and, optionally, the image and is responsible for producing a single best output. In the following, we explore in detail our choices for both stages.

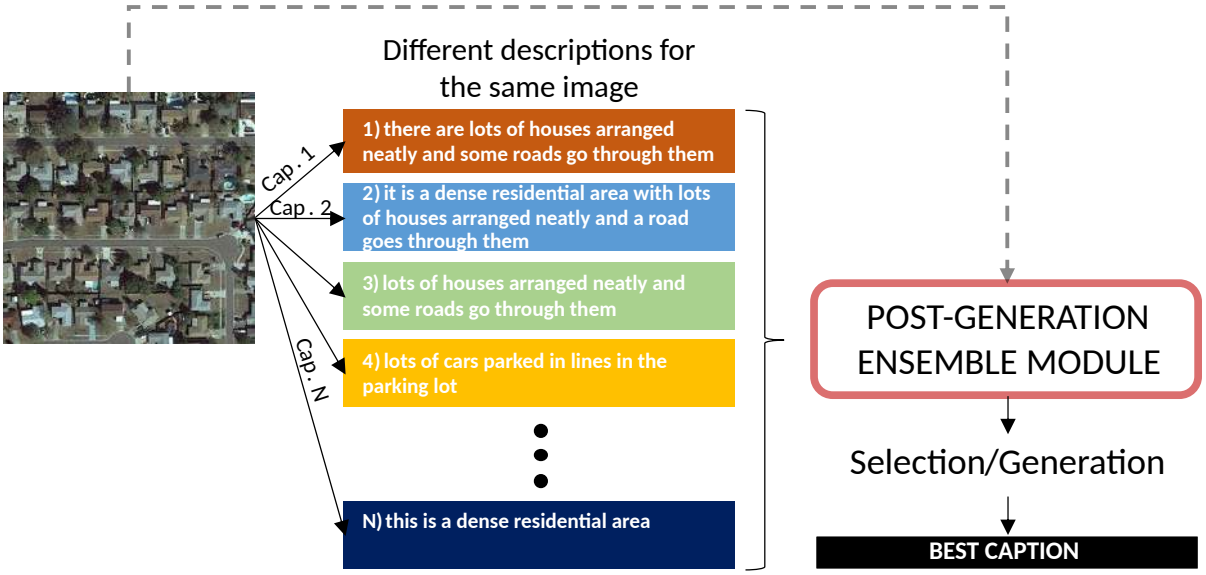


Figure 3.1: Conceptual pipeline of our ensemble approach. From an image, several captions are generated using different captioners. A post-generation ensemble module takes this set of captions and, optionally, the image, outputting a single best caption.

Caption generation stage

As stated in [81], the effectiveness of an ensemble largely depends on the diversity among its members. If all classifiers in the ensemble make the same errors, the ensemble will not perform significantly better than its individual members. Instead, if the members make errors in different instances, an ensemble can compensate for individual weaknesses. Diversity can be achieved employing different architectures, training recipes, and trainset distributions. In image captioning, since the output is a textual description, differences can also arise from the vocabulary used by the algorithm. We created an ensemble of $N = 7$ different models, including three state-of-the-art pre-trained captioners, namely MLAT [85], Blip-2 [86], and CapDec [87]. Further, we build and train four additional models following the architecture of [33]. We achieve architectural diversity using different combinations of encoder and decoder, as depicted in Table 3.2. We created models combining three encoders: two CNNs (VGG-16 and ResNet-50) and a Vision Transformer (ViT-B/16), and two decoders: Gated Recurrent Unit (GRU) and a Transformer Decoder (TD). It is worth mentioning that CapDec and Blip-2 are trained on natural scenes and, thus, are not tailored to the remote sensing scenario. Given a dataset $\mathcal{D} = \{(\mathbf{X_1}, \mathbf{T_1}), (\mathbf{X_2}, \mathbf{T_2}), \dots\}$ of samples of images and their corresponding description, the parameters of each custom model are adjusted to maximize the likelihood of the model generating the correct description for each image, as depicted in Equation 2.5. During inference, we generate the caption using greedy sampling to enforce reproducibility.

Post-generation fusion stage

We envisioned two primary post-processing strategies: selection and generation. Selection strategies prove advantageous in scenarios marked by pronounced uncertainty among captioning models, where a significant proportion of generated captions might lack relevance to the image content. Conversely, generation becomes pertinent when there is substantial semantic concordance among the candidates, but syntactical discrepancies or inaccuracies in the generated captions could compromise both their

Table 3.2: Algorithms included in our ensemble.

	Encoder	Decoder
CC-a	VGG-16	GRU
CC-b	ResNet-50	GRU
CC-c	ResNet-50	Transformer
CC-d	ViT	Transformer
MLAT [85]	ResNet-50	Transformer
Blip-2 [86]	ViT	FlanT5 XXL [88]
CapDec [87]	RN-50x4	GPT-2

CC stands for **C**ustom **C**aptioner.

performance and readability.

In the following, we introduce two selection strategies: naïve selection and CLIP-Coherence (CLIP-C) selection. Additionally, we introduce a generative strategy grounded in the Variational Autoencoder (VaE) framework.

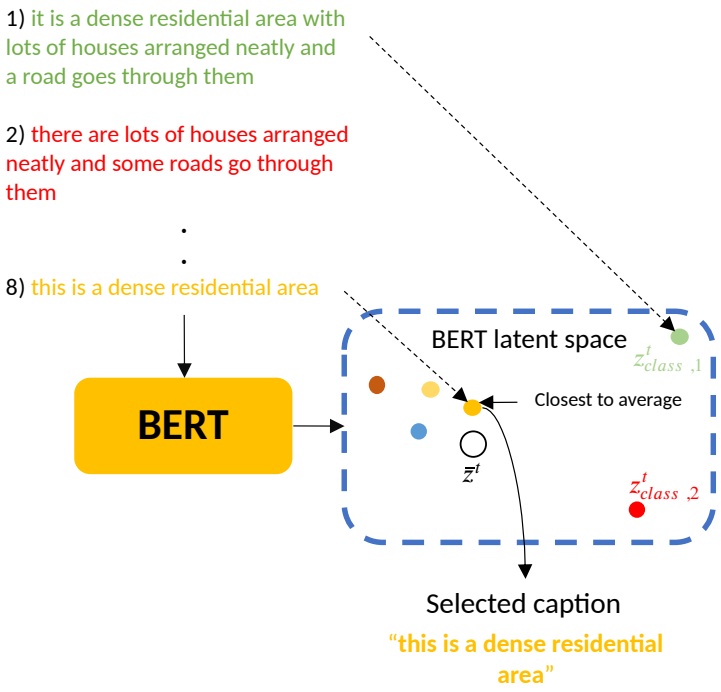


Figure 3.2: The naïve selection strategy. Each candidate caption generated in the first stage is projected into a semantic latent space using a pre-trained BERT [89] model. The average of the embeddings is used as an anchor, and the caption that falls closer to it is selected as the final output.

Naïve selection

Naive selection, depicted in Figure 3.2, is a text-only strategy in which the selection process relies solely on the generated set of captions. It leverages a distilled version of the BERT model (Bidirectional Encoder Representations from Transformers [89]), which is suited to convert texts into semantic vector representations. This operation is depicted in Equation 3.1.

$$\begin{aligned} \mathrm{Z}^t &= [z^t_{class}; z^t_1; z^t_2; \ldots; z^t_n] = \mathrm{BERT(T)}, \quad \mathrm{T} = [t_{class}; t_1; t_2; \ldots; t_n] \\ z^t_{class} &= \mathrm{Pooling}\big([z^t_{class}; z^t_1; z^t_2; \ldots; z^t_n]\big) \end{aligned} \tag{3.1}$$

Where $\mathrm{T} \in \mathbb{R}^{(n+1)}$ represent the sequence of text tokens obtained from the tokenization of a text $\mathbf{T}$, with a special "class" token prepended at the beginning and $\mathrm{Z}^t \in \mathbb{R}^{(n+1)\times h}$ is the sequence of output embeddings, which is the result of the processing of several self-attention layers inside the BERT model, and Pooling is the operation of selecting the output embedding corresponding to the "class" token. In their paper, the authors pre-train their model on a large corpus of text using two self-supervised objectives: masked language modeling prediction and next-sentence prediction. This massive pre-training allows their model to extract semantic sentence representations that can benefit a wide suite of tasks such as question-answering and language inference. The model projects semantically related sentences in similar parts of the latent space, creating a semantic latent space that can be leveraged for similarity search and other tasks. We adopted the distilled 6-layer version from [90] to decrease the computational requirements. Considering a set of M candidate captions, Naive selection proceeds as follows:

- **Projection**: each candidate caption is projected in the BERT latent space as detailed in Equation 3.1, resulting in a set of embeddings $\{z^t_{class,1}, z^t_{class,2}, \ldots, z^t_{class,M}\}$, where each one represents the content of one candidate caption.
- **Aggregation**: an anchor $\bar{z}^t$ is calculated as the average of all the sentence embeddings $\bar{z}^t = \frac{1}{M}\sum_{i=1}^{M} z^t_{class,i}$. The anchor serves as the prototype vector encapsulating the average semantic content of the set of captions.
- **Selection**: the caption whose embedding has the smallest Euclidean distance from the anchor is selected as the final caption.

Within the landscape of ensemble techniques, this approach shares similarities with the majority voting principle.

CLIP-C Selection

CLIP-C (CLIP-Coherence) selection, depicted in Figure 3.3, employs the Contrastive Language-Image Pre-training (CLIP) model [91], a multi-modal framework that integrates textual and visual data. CLIP consists of two branches, one processing the image and the other processing text, both generating embeddings following 3.1. The text branch is a Transformer encoder [43] while the visual branch is a Vision Transformer (ViT) [92]. Using a dataset of 400 M of corresponding image-text pairs, contrastive learning is used to enforce CLIP to project the two modalities in a shared latent space. Given an image $\mathbf{X}$, a *corresponding* text is a semantically coherent description of the image such as a caption or an anecdote about the image content. A text which is semantically unrelated to the image content is referred to as a mismatched text. After training using contrastive learning, the latent space shows an interesting property: corresponding images and texts are projected in similar regions, while mismatched pairs are projected far apart. Thanks to this property, CLIP has demonstrated robust capabilities in zero-shot classification and retrieval tasks on natural scenes, recently extending to remote sensing scenes [93][94][95]. The CLIP-Coherence selection employs CLIP to project the image and the candidate captions in the

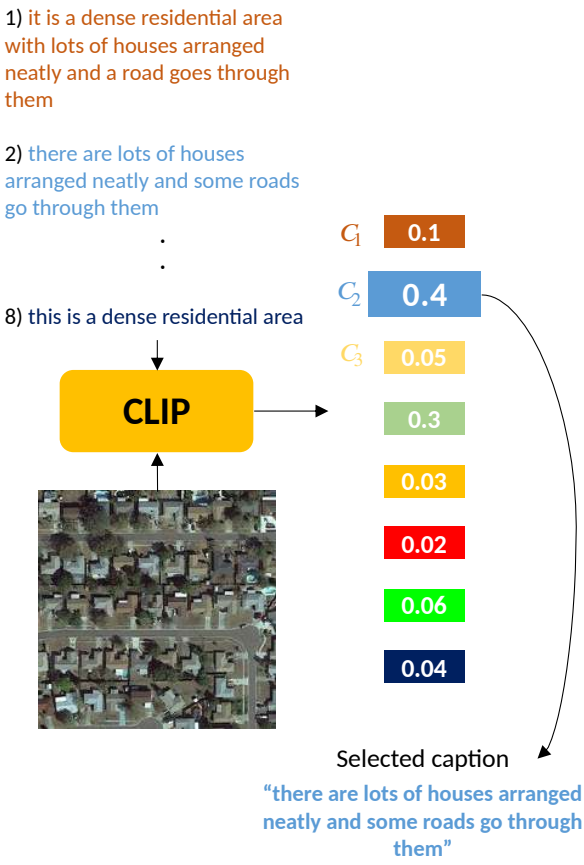


Figure 3.3: The CLIP-Coherence selection strategy. Using a pre-trained CLIP model, an image-text coherence score is derived for each candidate caption. The one with the highest coherence is selected as the final output.

shared latent space and evaluate the coherence of the image with each candidate via the cosine similarity between the respective embeddings:

$$\text{cosine similarity}(a, b) = \frac{a \cdot b}{\|a\|\|b\|} \tag{3.2}$$

Where $a \in R^d$ and $b \in R^d$ represent two generic vectors of dimension d and $\|\cdot\|$ denotes the Euclidean norm. We conducted experiments using two versions of the CLIP model. The first is the pre-trained version, released by OpenAI, and trained on a comprehensive corpus of web-scraped text-image pairs (mainly covering natural scenes). The second [93] is fine-tuned on remote-sensing scenes, thus exhibiting enhanced performance in tasks such as RS image retrieval and RS zero-shot classification. Supposing a set of M candidate captions and an RGB image $\mathbf{X} \in \mathbb{R}^{3\times w\times h}$, the CLIP-Coherence selection proceeds as follows:

- **Projection**: the text branch projects each candidate caption in the shared latent space, producing a set of embeddings $\{z^t_{class,1}, z^t_{class,2}, \ldots, z^t_{class,M}\}$. The visual branch projects the image in the same space, producing the embedding z^x_{class}.

- **Similarity computation**: the cosine similarity of each candidate caption with the image is computed as $C_i = \text{cosine similarity}(z^t_{class,i}, z^x_{class})$.
- **Selection**: the final output is the caption that maximizes the cosine similarity with the image $i^* = \arg\max_{i \in \{1,\dots,M\}} C_i$

From our perspective, the CLIP-coherence selection strategy facilitates using an ensemble of *specialized* members. Each member can excel in a distinct subset of scenes without needing a universal model across the entire spectrum of possible scenes. Notably, this procedure does not involve any averaging. Consequently, even in scenarios in which only a single caption exhibits coherence with the image, this strategy can, in principle, be able to isolate it, irrespective of the noise provided by the remaining candidates.

VaE fusion

Variational auto-Encoder (VaE) fusion, depicted in Figure 3.4, is a text-only strategy based on the Variational Auto-Encoder framework. This framework is preferred over a plain auto-encoder for the characteristics of the latent space achieved by noisy sampling during training. It has been demonstrated that textual interpolation in such latent space yields smoother and more plausible outputs [96] compared to standard auto-encoders. The idea is to compress the set of captions into the VaE latent space to preserve the semantic meaning of the set of captions while discarding possible syntactic errors. The noisy sampling used during training can enhance its robustness to slight perturbations of the inputs, making it the best candidate for dealing with such errors. However, this requires that we train a VaE model for each dataset. Due to data restriction, we decided to fine-tune a pre-trained VaE-based language model called OPTIMUS [97] instead of training a model from scratch. OPTIMUS is composed of two sub-networks: BERT [89] and GPT-2 [62]. As we previously saw, BERT acts as the encoder, condensing a text $\mathbf{T}$ into a latent representation z^t_{class}. The training of the VaE model follow these steps. The BERT representation $z^t_{class,i}$ of a text $\mathbf{T}$ is mapped via the linear projection f to the mean μ and standard deviation σ of a Gaussian probability distribution as:

$$\mu_i, \sigma_i = f(z^t_{class,i}) \tag{3.3}$$

A noisy latent vector is sampled from the Gaussian distribution as:

$$\tilde{z}^t_{class,i} = \mu_i + \sigma_i \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{3.4}$$

where ϵ is drawn from a standard normal distribution. The text is reconstructed by GPT-2 conditioned on the noisy latent vector

$$\hat{\mathbf{T}} = \text{GPT-2}(\tilde{z}^t_{class,i}) \tag{3.5}$$

where $\hat{\mathbf{T}}$ is the reconstructed sentence. The VaE model is trained to minimize the following loss function:

$$\mathcal{L}_{vae} = \mathbb{E}_{q(z|\mathbf{T})}\left[\log p(\mathbf{T}|z)\right] - D_{KL}\big(q(z|\mathbf{T}) \| p(z)\big) \tag{3.6}$$

where $\mathbb{E}_{q(z|\mathbf{T})}\left[\log p(\mathbf{T}|z)\right]$ is the likelihood of reconstructing the original text $\mathbf{T}$ from its latent representation z, averaged over the latent posterior distribution $q(z|\mathbf{T})$. $p(z)$ is the prior latent distribution, usually a standard normal distribution $\mathcal{N}(0, 1)$, and D_{KL} is the Kullbach-Leibler (KL) divergence, used to push $q(z|\mathbf{T})$ close to $p(z)$, smoothing the latent space and preventing the network to overfit on individual samples.

OPTIMUS has been pre-trained on roughly $2M$ sentences from English Wikipedia. The authors used pre-processing to isolate sentences of a maximum length of 64 tokens to make it more targeted to modeling short sentences. Starting from the pre-trained weights, we fine-tune 4 versions of OPTIMUS, one for each of the 4 datasets. After fine-tuning of the OPTIMUS model, supposing a set of M candidate captions, the VaE fusion strategy proceeds as follows:

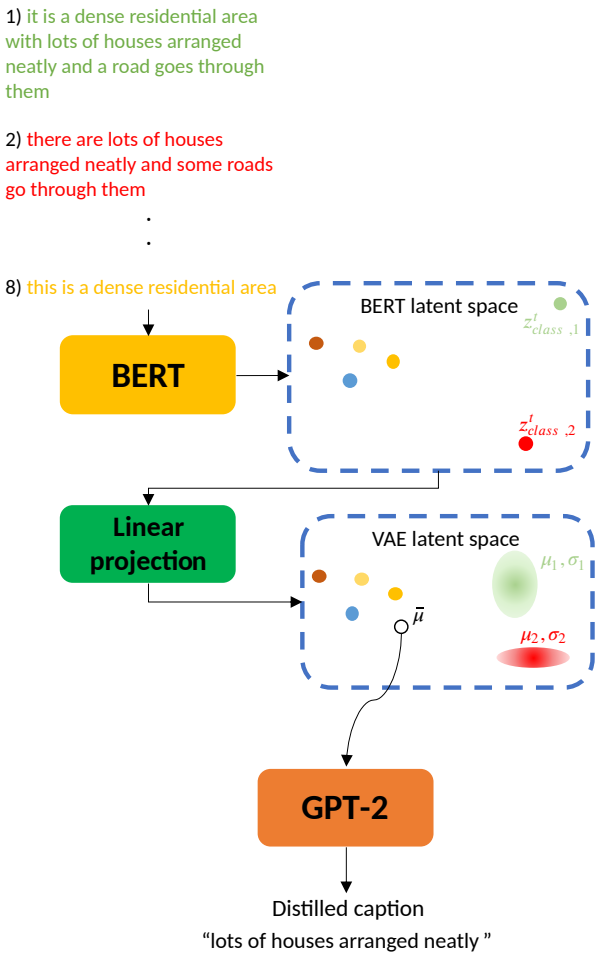


Figure 3.4: The VaE fusion strategy. Inside OPTIMUS, the encoder (BERT) projects the candidate captions in a smooth VAE latent space. There, the Gaussians means are averaged, and used as anchor to condition the decoder (GPT-2), which distills the final caption.

- **Projection**: Each candidate caption is encoded into the VaE latent space. This process produces two sets of M elements: the mean values $\{\mu_1, \mu_2, \ldots, \mu_M\}$ and standard deviations $\{\sigma_1, \sigma_2, \ldots, \sigma_M\}$, of M Gaussian distributions.
- **Aggregation**: The aggregated latent representation is obtained as the average of each Gaussian's mean, $\bar{\mu} = \frac{1}{M}\sum_{i=1}^{M} \mu_i$.
- **Reconstruction**: the representation $\hat{z}^t_{class,i} = \bar{\mu}$ conditions the OPTIMUS decoder to generate the distilled caption.

The idea is to leverage the modeling capability of the variational latent space to retain a condensed representation of the input captions, thus discarding noise from possible errors or misspellings. Decoding from $\hat{z}^t_{class,i} = \bar{\mu}$ can be seen as extracting the condensed semantic meaning of the input set of captions. We formulate this strategy to deal with possible syntactic errors in the input captions. Indeed, such

a scenario cannot be tackled using selective strategies, which are restricted to the pool of generated candidates. Details of architecture and training are provided in Section 3.1.2.

3.1.2 Experimental results

Setup

As our primary focus is not on the performance of individual algorithms, we do not search for the best training hyper-parameters. Instead, we apply the same hyper-parameters across all custom captioners, selecting reasonable values based on the data volume at our disposal. Each captioner uses the BERT tokenizer with an embedding dimension of $d_t = 256$. For training, we used a batch size of $bs = 8$, a learning rate of $\alpha = 1 \times 10^{-4}$, and a dropout probability of $p_{drop} = 0.15$ to counteract over-fitting. During training, we freeze the parameters of the encoder while updating the parameters of the decoder. Due to the limited dataset size, we use early stopping monitoring the BLEU-4 on the validation set. The AdamW optimizer was employed with a weight decay of $\lambda = 1 \times 10^{-7}$.

OPTIMUS has been fine-tuned for $e = 10$ epochs, adhering to the pre-training configuration provided by the original authors.

The performance is reported using BLEU1-4 (B1-4), ROUGE-L (R), METEOR (M), and CIDEr (C). In the following, we present three evaluation scenarios to analyze the ensemble's performance under different conditions.

Scenario 1: Standard Evaluation

This experiment reproduces a standard scenario in which evaluation is conducted on samples from the same dataset as the one used in training (of course, of separate train and test subsets). In each table, the first seven rows represent the performance of the single captioners, while the last four are the ensemble results with the three strategies. Color codes are used to indicate RS datasets used to train the captioners. Blip-2 and CapDec have been trained on datasets of natural scenes.

Table 3.3: UCM-Captions: Standard Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-a	0.79	0.72	0.66	0.61	0.74	0.40	3.03
CC-b	0.77	0.70	0.64	0.59	0.71	0.39	2.99
CC-c	0.54	0.45	0.38	0.34	0.48	0.22	1.56
CC-d	**0.82**	**0.75**	**0.70**	**0.65**	**0.77**	**0.43**	**3.19**
MLAT [85]	0.42	0.23	0.13	0.09	0.31	0.14	0.54
Blip-2 [86]	0.35	0.19	0.10	0.04	0.27	0.13	0.33
CapDec [87]	0.30	0.16	0.08	0.04	0.26	0.11	0.18
Strategy							
Naïve	0.80	0.72	0.66	0.60	0.74	0.40	3.09
CLIP-rsicdv2	0.69	0.61	0.56	0.52	0.62	0.34	2.49
CLIP-vitlarge14	0.55	0.44	0.38	0.34	0.45	0.24	1.48
VaE	0.76	0.67	0.60	0.54	0.70	0.37	2.69

Trained on UCM
Trained on RSICD

Table 3.4: UAV-Captions: Standard Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-a	0.69	0.59	0.49	0.40	0.70	0.34	3.92
CC-b	0.70	0.59	0.48	0.38	0.69	0.34	3.76
CC-c	0.59	0.48	0.39	0.31	0.60	0.27	3.08
CC-d	0.68	0.57	0.46	0.37	0.69	**0.35**	3.56
MLAT [85]	0.13	0.03	0.01	0.00	0.13	0.06	0.04
Blip-2 [86]	0.21	0.07	0.03	0.00	0.18	0.10	0.11
CapDec [87]	0.13	0.05	0.01	0.00	0.13	0.09	0.04
Strategy							
Naïve	**0.73**	**0.62**	**0.52**	**0.42**	**0.72**	**0.35**	**4.00**
CLIP-rsicdv2	0.35	0.24	0.18	0.13	0.35	0.18	1.08
CLIP-vitlarge14	0.27	0.16	0.11	0.07	0.25	0.16	0.52
VaE	0.68	0.57	0.45	0.35	0.66	0.31	3.28

Trained on UAV
Trained on RSICD

Table 3.5: SIDNEY-Captions: Standard Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-a	**0.77**	**0.68**	**0.61**	**0.55**	**0.70**	**0.38**	**2.35**
CC-b	0.73	0.63	0.56	0.49	0.66	0.35	2.11
CC-c	0.69	0.60	0.52	0.45	0.64	0.34	1.91
CC-d	0.76	0.67	0.60	0.53	**0.70**	**0.38**	2.31
MLAT [85]	0.47	0.24	0.13	0.07	0.29	0.15	0.24
Blip-2 [86]	0.33	0.18	0.12	0.07	0.28	0.11	0.10
CapDec [87]	0.31	0.11	0.00	0.00	0.26	0.08	0.08
Strategy							
Naïve	0.76	0.67	0.60	0.53	**0.70**	**0.38**	2.34
CLIP-rsicdv2	0.59	0.46	0.39	0.34	0.50	0.25	1.58
CLIP-vitlarge14	0.56	0.44	0.37	0.33	0.46	0.23	1.11
VaE	0.71	0.61	0.54	0.47	0.65	0.35	1.60

Trained on SIDNEY
Trained on RSICD

The ensemble's performance is summarized in Tables 3.3-3.6. Most captioners achieve satisfactory results, with the notable exceptions of Blip-2 and CapDec, which lag behind across all evaluation metrics. It is important to highlight that these models have not been specifically trained on remote sensing data, making their lower scores somewhat expected. Furthermore, as we can see from the performance of MLAT, a simple change in the training dataset can be critical when evaluating the model using standard metrics. MLAT has been trained on the RSICD dataset, and we can see that on UCM, UAV, and SIDNEY, it achieves lower metrics than the first four models trained on the same dataset. Despite this,

Table 3.6: RSICD-Captions: Standard Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-a	0.61	0.44	0.34	0.27	0.46	0.24	0.71
CC-b	0.60	0.43	0.32	0.26	0.44	0.24	0.69
CC-c	0.54	0.36	0.26	0.20	0.38	0.20	0.48
CC-d	0.63	0.47	0.37	0.30	0.48	0.26	0.81
MLAT [85]	**0.65**	**0.49**	**0.39**	**0.32**	0.49	**0.27**	**0.90**
Blip-2 [86]	0.34	0.16	0.08	0.04	0.24	0.10	0.20
CapDec [87]	0.35	0.16	0.07	0.03	0.23	0.10	0.13
Strategy							
Naïve	**0.65**	**0.49**	**0.39**	**0.32**	**0.50**	**0.27**	0.86
CLIP-rsicdv2	0.58	0.42	0.32	0.26	0.43	0.23	0.73
CLIP-vitlarge14	0.46	0.28	0.19	0.15	0.31	0.15	0.40
VaE	0.63	0.44	0.33	0.26	0.46	0.24	0.73

Trained on RSICD

the Naive selection approach performs consistently well, often aligning with or exceeding the best results from individual captioners (Table 3.4). Moreover, the Variational Autoencoder framework demonstrates that it can effectively capture the overall meaning of the set of captions by achieving a satisfactory performance, often higher than most of the captioners in the ensemble. Unexpectedly, the CLIP selection pipeline hinders performance, particularly when using CLIP-vitlarge14, which is not fine-tuned for the remote sensing domain. Two factors may contribute to this outcome. First, the CLIP model is not particularly specialized for RS image captioning, and second, it often selects captions produced by Blip-2, which are generally coherent, although arranged using a different word distribution, impacting on the scores. These observations indicate a need to reconsider the evaluation metrics used in image captioning. Current metrics, which prioritize syntax and exact word matching over semantic integrity, may not fully capture the quality of a generated caption. This observation is confirmed in the qualitative results depicted in Figures 3.5 and 3.6. It can be noticed how the ensemble, and especially the selective strategies, can often select a very coherent caption.

Scenario 2: Generalization evaluation

This scenario simulates a higher diversity between training and testing data. Each dataset, in turn, undergoes prediction using captioners that have been trained on every other dataset, thus excluding the dataset in focus. This design aims to test the generalization capabilities of the algorithms when exposed to unfamiliar data and the possible benefits of using the ensemble in such a scenario. For clarity, we keep only the best-scoring custom captioner for each dataset.

In Tables 3.7-3.10, a marked decline in the performance of all captioners is observed across every metric and dataset. This is especially true for models trained on the UAV-Captions dataset, which consistently shows the lowest performance metrics, as highlighted by the yellow lines in the tables. Just by looking at the metrics, we can deduce that UCM, RSICD, and SYDNEY datasets share certain features or characteristics that make them more closely aligned, while UAV is substantially different. Among ensemble approaches, the Variational Autoencoder (VaE) fusion is the most effective method. The unique strength of this approach lies in the capability of the VaE to act as a semantic 'translator.'

Table 3.7: RSICD-Captions: Generalization Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-d	0.38	**0.18**	**0.09**	**0.05**	**0.24**	**0.10**	0.14
CC-d	0.33	0.13	0.05	0.02	0.21	**0.10**	0.08
CC-d	0.20	0.09	0.02	0.00	0.18	0.05	0.02
Blip-2 [86]	0.33	0.16	0.07	0.04	0.23	**0.10**	0.19
CapDec [87]	0.35	0.15	0.07	0.03	0.23	**0.10**	0.11
Strategy							
Naïve	0.36	0.16	0.07	0.03	0.22	**0.10**	0.14
CLIP-rsicdv2	0.35	0.17	0.08	0.04	**0.24**	**0.10**	**0.21**
VaE	**0.40**	0.17	0.08	0.03	**0.24**	**0.10**	0.10

Trained on UCM
Trained on UAV
Trained on SIDNEY

Table 3.8: SIDNEY-Captions: Generalization Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-d	0.53	0.41	0.32	**0.27**	0.42	0.20	0.44
CC-d	0.47	0.22	0.12	0.07	0.29	0.16	0.27
CC-b	0.16	0.08	0.00	0.00	0.14	0.04	0.04
MLAT [85]	0.49	0.25	0.13	0.08	0.30	0.16	0.28
Blip-2 [86]	0.31	0.18	0.12	0.00	0.28	0.10	0.10
CapDec [87]	0.28	0.10	0.04	0.00	0.22	0.07	0.07
Strategy							
Naïve	0.49	0.26	0.14	0.08	0.32	0.16	0.25
CLIP	0.41	0.24	0.15	0.06	0.31	0.14	0.27
VaE	**0.64**	**0.46**	**0.33**	0.24	**0.47**	**0.24**	**0.86**

Trained on UCM
Trained on UAV
Trained on RSICD

The VaE is fine-tuned on the captions of the target dataset, allowing it to adapt its decoder to the specific vocabulary. This adaption facilitates a sort of semantic distillation through the latent space, thereby 'translating' the global meaning of the set of captions into the language style of the target dataset. This improves the relevance of the captions generated, resulting in a notable improvement across all performance metrics. The VaE is thus able to focus on the semantic aspects of input captions while overlooking syntactic variations or variations in the choice of words. Unlike other selective ensemble methods, the VaE inherently performs this semantic translation, making it a useful tool for bridging the semantic gap across diverse training datasets. Qualitative results, depicted in Figures 3.7 and 3.8, highlight a difficult scenario for the ensemble, in which most of the captions are unrelated and not coherent with the image. The most robust alternative in this case is the CLIP-selection strategy, particularly CLIP-rsicdv2, which provides coherent captions for all the images. Its general, non-specialized counterpart, CLIP-vitlarge14, is tricked in the first image on the UCM-Captions dataset but provides coherent captions

Table 3.9: UAV-Captions: Generalization Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-c	0.21	0.12	0.03	0.01	0.23	0.08	0.05
CC-b	0.11	0.03	0.01	0.00	0.10	0.05	0.04
CC-d	0.19	0.09	0.03	0.00	0.20	0.07	0.05
MLAT [85]	0.12	0.02	0.00	0.00	0.13	0.05	0.04
Blip-2 [86]	0.20	0.08	0.03	0.02	0.19	0.11	0.11
CapDec [87]	0.14	0.06	0.02	0.01	0.13	0.09	0.05
Strategy							
Naïve	0.14	0.05	0.01	0.00	0.14	0.06	0.05
CLIP	0.19	0.08	0.03	0.02	0.18	0.10	0.10
VaE	**0.37**	**0.24**	**0.15**	**0.09**	**0.34**	**0.15**	**0.37**

Trained on UCM
Trained on SIDNEY
Trained on RSICD

Table 3.10: UCM-Captions: Generalization Evaluation. Bold entries highlight the best results.

Captioner	**B1**	**B2**	**B3**	**B4**	**R**	**M**	**C**
CC-b	0.22	0.11	0.00	0.00	0.19	0.06	0.04
CC-d	0.42	0.20	0.10	0.05	0.28	0.13	0.38
CC-d	0.37	0.22	0.13	0.07	0.32	0.15	0.29
MLAT [85]	0.42	0.23	0.13	0.08	0.31	0.14	0.54
Blip-2 [86]	0.34	0.20	0.12	0.08	0.29	0.13	0.40
CapDec [87]	0.31	0.15	0.06	0.02	0.25	0.11	0.16
Strategy							
Naïve	0.43	0.24	0.14	0.09	0.32	**0.16**	0.47
CLIP	0.40	0.24	0.14	0.08	0.32	0.15	0.49
VaE	**0.47**	**0.32**	**0.22**	**0.16**	**0.38**	**0.16**	**0.62**

Trained on UAV
Trained on SIDNEY
Trained on RSICD

for all the other cases.

Scenario 3: Robustness Evaluation

In this setup, we want to test the resilience of the ensemble to errors, such as semantic errors, syntactic errors, or misspellings. We consider different noise levels, corresponding to different percentages of corrupted words over the total *word count* in the input set of captions. The corruption level is represented as:

$$\text{Corruption Level} = \frac{\text{Number of Corrupted Words}}{\text{Total Word Count}} \times 100 \tag{3.7}$$

Table 3.11: SIDNEY-Captions: Robustness Evaluation. Results Expressed in Terms of Bleu-4. Bold entries highlight the best results.

Noise level (%)	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Best captioner	**0.55**	0.50	0.44	0.40	0.34	0.30	0.26
Worst captioner	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Naïve	0.53	**0.53**	**0.49**	**0.45**	0.40	0.35	0.29
CLIP-rsicdv2	0.34	0.33	0.29	0.27	0.26	0.24	0.20
CLIP-vitlarge14	0.33	0.32	0.33	0.30	0.26	0.27	0.20
VaE	0.46	0.46	0.46	**0.45**	**0.46**	**0.44**	**0.41**

Table 3.12: RSICD-Captions: Robustness Evaluation. Results Expressed in Terms of Bleu-4. Bold entries highlight the best results.

Noise level (%)	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Best captioner	**0.32**	0.29	0.25	0.22	0.19	0.17	0.14
Worst captioner	0.02	0.02	0.02	0.02	0.02	0.01	0.01
Naïve	**0.32**	**0.31**	**0.28**	**0.26**	**0.23**	**0.20**	**0.17**
CLIP-rsicdv2	0.26	0.24	0.22	0.20	0.18	0.16	0.14
CLIP-vitlarge14	0.14	0.14	0.14	0.13	0.12	0.11	0.10
VaE	0.26	0.25	0.22	0.19	0.14	0.11	0.08

Errors are simulated using word deletion, word substitution, and character substitution. We evaluate the ensemble's performance across seven noise levels, ranging from 0% to 30%, incremented in steps of 5%.

Results are provided in Tables 3.11-3.14. A key observation is the different performance of selective and generative strategies under varying levels of noise corruption. Specifically, under low noise conditions, selective strategies exhibit superior performance. In contrast, under conditions of elevated noise, the Variational Autoencoder (VaE) generative approach demonstrates a slower decrease in performance. This trend is not observed for the RSICD-Captions dataset. We hypothesize that the bigger size of this dataset increases the variability of captions and scenes, thereby making it more difficult for the VaE to adapt to its distribution. In contrast, smaller datasets present a lower range of variance and complexity, which may prove the VaE more effective in adapting to the dataset's caption distribution. Importantly, our experiments validate the efficacy of ensemble strategies in mitigating the impact of syntactic errors. This suggests that ensemble methods can serve as a robust countermeasure against various forms of linguistic noise, thereby enhancing overall resilience. Qualitative results are depicted in Figures 3.9 and 3.10.

3.1.3 Discussion

After collecting and analyzing all the results for the three proposed configurations, we summarize our findings and insights on applying a post-generation ensemble for image captioning. As the results demonstrate, the ensemble can increase the robustness of the output to various situations, scenes, vocabulary, and other factors of variation. Upon the proposed techniques, the main strengths and drawbacks are reported in Table 3.15.

Table 3.13: UCM-Captions: Robustness Evaluation. Results Expressed in Terms of Bleu-4. Bold entries highlight the best results.

Noise level (%)	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Best captioner	**0.65**	**0.59**	0.52	0.46	0.40	0.36	0.30
Worst captioner	0.04	0.03	0.03	0.02	0.02	0.02	0.02
Naïve	0.60	**0.59**	**0.54**	0.50	0.44	0.38	0.33
CLIP-rsicdv2	0.52	0.48	0.44	0.40	0.37	0.34	0.28
CLIP-vitlarge14	0.34	0.35	0.34	0.33	0.30	0.28	0.24
VaE	0.54	0.52	0.52	**0.51**	**0.47**	**0.45**	**0.42**

Table 3.14: UAV-Captions: Robustness Evaluation. Results Expressed in Terms of Bleu-4. Bold entries highlight the best results.

Noise level (%)	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Best captioner	0.40	0.37	0.33	0.29	0.24	0.21	0.18
Worst captioner	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Naïve	**0.42**	**0.41**	**0.39**	**0.37**	**0.33**	**0.29**	**0.26**
CLIP-rsicdv2	0.13	0.13	0.13	0.13	0.12	0.12	0.13
CLIP-vitlarge14	0.07	0.08	0.08	0.09	0.09	0.09	0.09
VaE	0.36	0.35	0.34	0.32	0.31	**0.29**	**0.26**

Despite its simplicity, the naïve selection approach proves to be a very strong baseline. Its main problem is the lack of prior filtering on the set of input captions, which also affects the VaE fusion strategy. In our opinion, the most promising and scalable approach is the CLIP-C selection, which provides an effective way of dealing with situations of high variability in the input set. We speculate that the release of more targeted CLIP models for the RS field can further improve the selection ability and, thus, the robustness of the CLIP-C selection strategy. Furthermore, we analyzed the additional computation time required to run our ensemble. This mainly depends on the number of models included in the ensemble, with little additional computation time required by the selection/fusion methods. Table 3.16 reports the time used by each captioner to produce a caption for an image and the additional computational time for the fusion strategy, along with the total time. As can be seen, running the ensemble requires the generation of a caption by each model plus the added time for the fusion strategy. This leads to a severe increase in computational time, with the slowest strategy being VaE fusion, taking an average of 4.045 seconds to process an image.

3.1.4 Conclusion

We have systematically investigated the possibility of applying an ensemble approach to increase the robustness of image captioning. Specifically, we designed strategies that act a-posteriori after individual captions have been generated. Two *selective* strategies, Naïve and CLIP-selection, focus on choosing the most coherent caption from the set of candidates. A *generative* strategy, based on the variational autoencoder (VaE) framework, synthesizes a new caption based on the entire set of candidates. We

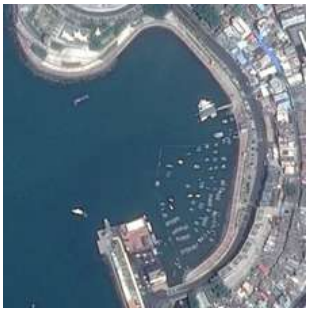
CC-a - this is a beach with blue sea and white sands
CC-b - the waves slapping a white sand beach
CC-c - this is a dense forest with green waters and grass
CC-d - the waves slapping a white sand beach over and over again
MLAT - a white waves is near a yellow beach
Blip2 - a man is walking on the beach
CapDec - A woman in a white dress is standing in the snow.
Naïve - a white waves is near a yellow beach
CLIP-rsicdv2 - the waves slapping a white sand beach over and over again
CLIP-vitlarge14 - a man is walking on the beach
VaE - this is a beach with white sands and blue waters

CC-a - it is a straight runway with some mark lines on it
CC-b - there are two straight freeways closed to each other
CC-c - there are some mark lines on the straight runway
CC-d - there are four airplanes in the airport
MLAT - many planes are parked in an airport
Blip2 - aerial view of a small airport
CapDec - A group of planes parked in an airport.
Naïve - A group of planes parked in an airport.
CLIP-rsicdv2 - A group of planes parked in an airport.
CLIP-vitlarge14 - A group of planes parked in an airport.
VaE - there are two airplanes with black fuselage taxiing on the runway

Figure 3.5: Qualitative Results on UCM-Captions, scenario 1.

Table 3.15: Comparative Analysis of Captioning Strategies

Strategy	**Strengths**	**Drawbacks**
Naïve	• Easy to integrate • Good trade-off between robustness and computational demand	• Unstable with high caption semantic variance • Ignores coherence of the captions during semantic "averaging" in the latent space • Cannot handle syntactic errors
CLIP-C	• Easy to integrate • Stable with high semantic variance • Selective behavior for image coherence	• Cannot handle syntactic errors • Not tailored for remote sensing images
VaE	• Handles syntactic errors	• Ignores coherence of the captions during semantic "averaging" in the latent space • Nondeterministic output due to generation sampling • Higher computational load



CC-a - a large number of trees were planted around the stadium	CC-a - a baseball field is near several green trees
CC-b - many boats are in a port near many buildings and green trees	CC-b - a baseball field is surrounded by many green trees
CC-c - many boats are in a port near many buildings	CC-c - a playground is surrounded by many green trees and buildings
CC-d - many buildings are near a port near many buildings	CC-d - a baseball field is surrounded by some green trees and buildings
MLAT - several boats are in a port near some buildings	MLAT - a baseball field is near several green trees
Blip2 - the harbor of hong kong	Blip2 - aerial view of a baseball field
CapDec - A large body of water with a boat and people on it.	CapDec - A baseball field with a baseball player holding a bat on it.
Naïve - several boats are in a port near some buildings	Naïve - a baseball field is near several green trees
CLIP-rsicdv2 - many buildings are near a port near many buildings	CLIP-rsicdv2 - a baseball field is surrounded by some green trees and buildings
CLIP-vitlarge14 - several boats are in a port near some buildings	CLIP-vitlarge14 - a baseball field is near several green trees
VaE - many ships are in a port near a beach	VaE - a baseball field is surrounded by many green trees

Figure 3.6: Qualitative Results on RSICD-Captions, scenario 1.

Table 3.16: Caption Generation Times and Ensemble Fusion Overhead (in seconds per image)

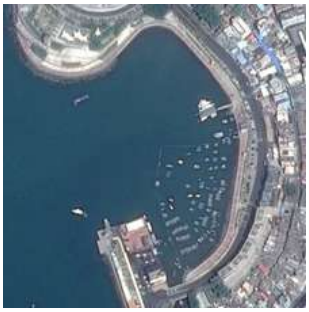
Model	**Generation**	**Total Time**	
CC-a	0.025		
CC-b	0.030		
CC-c	0.126		
CC-d	0.140	3.795	
MLAT	2.660		
Blip-2	0.654		
CapDec	0.160		
Ensemble	**Ensemble time**	**Total Time**	**% of Total Time**
Naive	0.010	3.805	0.26%
CLIP-C	0.038	3.833	0.99%
VAE	0.250	4.045	6.18%

conducted a comprehensive analysis using three well-established remote sensing image captioning datasets in addition to a novel dataset named UAV-Captions. Three scenarios have been designed to test different aspects of the captioning process, which allowed us to expose and discuss the strengths and weaknesses of each strategy. Our findings demonstrate that our ensemble-based approaches offer a scalable and robust way of integrating various captioning algorithms. This leads to more reliable and contextually accurate

CC-a - there is an asphalted zone	CC-a- there is large road
CC-b - there is soil ground	CC-b- there is white roof
CC-c - there is grass field	CC-c- grass field at upper right is close to asphalt
CC-d - there is large road	CC-d- there is parking lot
CC-a - a piece of ocean is near a yellow beach	CC-a - many planes are parked in an airport
CC-b - it is a piece of yellow desert	CC-b - some boats are in a large port near a road
CC-c - many green trees are in two sides of a curved river	CC-c - several storage tanks are near a piece of green meadow
CC-d - a piece of ocean is near a piece of green ocean	CC-d - many planes are parked in an airport
CC-a - a meadow with some green bushes and white bunkers on it while a highway passed by	CC-a - a small river with dark green waters goes through a residential area
CC-b - a meadow with some green bushes and white bunkers on it	CC-b - a part of ocean with deep green waters
CC-c - there are some white buildings in the industrial area with some roads go through	CC-c - there are some marking lines on the straight runway while some lawns beside
CC-d - a big meadow with some mark lines on it while a highway beside	CC-d - there are some white airplanes parked on the airport with some airport buildings beside
MLAT - a white waves is near a yellow beach	MLAT - many planes are parked in an airport
Blip2 - a man is walking on the beach	Blip2 - a small plane parked on the tarmac
CapDec - a close up of a person wearing a suit and tie	CapDec - A group of planes sitting on top of an airport tarmac.
Naïve - a meadow with some green bushes and white bunkers on it while a highway passed by	Naïve - many planes are parked in an airport
CLIP-rsicdv2 - a white waves is near a yellow beach	CLIP-rsicdv2 - A group of planes sitting on top of an airport tarmac.
CLIP-vitlarge14 - a man is walking on the beach	CLIP-vitlarge14 - A group of planes sitting on top of an airport tarmac.
VaE - there is a white sand beach with white sands and green waters	VaE - there are two tennis courts on the green ground with a road beside

Figure 3.7: Qualitative Results on UCM-Captions, scenario 2.

captions. More specifically, our results suggest that the CLIP coherence selection is less sensible to noisy and unrelated captions and, thus, more suitable in situations in which there is high semantic variability in the candidates. The approach based on the VaE framework has shown to be robust to noise, but, at low noise levels, it is outperformed by the selective strategies. In addition, we show that the use of multiple captioners incurs a significant computational overhead with respect to single-model alternatives, making the use of the ensemble suitable only when time is not a constraint. Avenues for future research in this area include the integration of better performing Contrastive Language-Image Pretraining (CLIP) models for remote sensing imagery [94], as well as the implementation of automated filtering mechanisms to prune less coherent caption candidates before ensemble application. The integration of tailored CLIP models can reduce the leaking of irrelevant captions, as happened in Figure 3.9 for the CLIP-vitlarge14. This model is trained on natural images, and we can see that from the colors in the image, the model is tricked into describing the image as a woman in a white dress. On the other hand, a mechanism to filter the captions before the Naive and VaE solutions can benefit both strategies by removing a portion



CC-a - there is an asphalted zone	CC-a- there is road
CC-b - there is an asphalted zone	CC-b- large white roof and some grass on the top
CC-c - there are several trees	CC-c- large white roof with shadow on the bottom right
CC-d - there are several rocks on the asphalt	CC-d- there is an asphalted zone
CC-a - there are two storage tanks on the ground	CC-a - it is a small baseball diamond with sand and grass
CC-b - lots of boats docked in lines at the harbor and the water is deep blue	CC-b - it is a small baseball diamond with sand and grass
CC-c - lots of cars parked in lines in the parking lot	CC-c - a medium residential area with a road goes through this area
CC-d - a big intersection with sky blue roofs	CC-d - it is a small baseball diamond
CC-a - a meadow with some green bushes and white bunkers on it while a highway passed by	CC-a - a residential area with houses arranged neatly and some roads go through this area
CC-b - a meadow with some green bushes and white bunkers on it	CC-b - a curved river with dark green waters goes through a residential area
CC-c - there are some white buildings in the industrial area with some roads go through	CC-c - there are some white marking lines on the runways while some lawns beside
CC-d - there are some sandlands and orange roofs arranged neatly	CC-d - there are some white buildings in the residential area with some roads go through
Blip2 - aerial view of a harbor with boats	Blip2 - a baseball field with a soccer field and a pool
CapDec - a number of small boats in a large body of water	CapDec - A baseball player holding a bat on top of a baseball field.
Naïve - there are some sandlands and orange roofs arranged neatly	Naïve - it is a small baseball diamond with sand and grass
CLIP-rsicdv2 - aerial view of a harbor with boats	CLIP-rsicdv2 - a baseball field with a soccer field and a pool
CLIP-vitlarge14 - aerial view of a harbor with boats	CLIP-vitlarge14 - a baseball field with a soccer field and a pool
VaE - there are some cement roads on the desert	VaE - a bareland with some cars is next to the straight roads

Figure 3.8: Qualitative Results on RSICD-Captions, scenario 2.

of irrelevant captions that could contaminate the selection and distillation with unrelated information. In summary, the ensemble strategies examined in this paper hold promise for significantly enhancing the reliability and contextual relevance of image captions in remote sensing applications.

CC-a - this is donnell beach blue sea and white sands
CC-b - the waves slapping a white sand beach
CC-c - this is g dense forest with green waters and grass
CC-d - the waves slapping a white sand beach over and over again
MLAT - a buyers deemed is near a yellow beach
Blip2 - a undercover is walking food the beach
CapDec - woman in a white dress is standing ip the snow.
Naïve - this is donnell beach blue sea and white sands
CLIP-rsicdv2 - the waves slapping a white sand beach over and over again
CLIP-vitlarge14 - woman in a white dress is standing ip the snow.
VaE - a white sand beach is taxing on one side and a small white sand beach on the other side

CC-a - it is straight runway with some on it
CC-b - there are straight freeways closen to each other
CC-c - there are some mark juarez on the straight runway
CC-d - there are airplanes in the airport
MLAT - many planes are parked in an airport
Blip2 - aerial viev of a small airport
CapDec - A group of planes parked in ancestral airport.
Naïve - there are airplanes in the airport
CLIP-rsicdv2 - A group of planes parked in ancestral airport.
CLIP-vitlarge14 - A group of planes parked in ancestral airport.
VaE - there are some airplanes with black fuselage taxiing on the runway

Figure 3.9: Qualitative Results on UCM-Captions, scenario 3. Noise level: 20%.

CC-a - a large number of quantitative were planted aroudd the stadium
CC-b - many boats are in a port near many buildings and khz hrees
CC-c - many boats are in n fort near many buildings
CC-d - many buildings are near a port near many budldings
MLAT - several boats are in a port near soqe buildings
Blip2 - the harbor of hong impulse
CapDec - A restaurant of water with i boat and people on ig.
Naïve - several boats are in a port near soqe buildings
CLIP-rsicdv2 - many buildings are near a port near many budldings
CLIP-vitlarge14 - the harbor of hong impulse
VaE - many ships in port are orderly surrounded by a port

CC-a - a 130 field is surrounded by some green trees and buildings
CC-b - a field fs stumble by many green trees
CC-c - o is surrounded by green trees and buildings
CC-d - a baseball field is surrounded by some green trees and buildings
MLAT - a baseball field is near several trees
Blip2 - aerial view of a baseball field
CapDec - o baseball field with a lo player holding a bat on it.
Naïve - a baseball field is near several trees
CLIP-rsicdv2 - l baseball field is near several breen trees
CLIP-vitlarge14 - aerial view of a baseball field
VaE - a baseball field locates in the meadow with several trees

Figure 3.10: Qualitative Results on RSICD-Captions, scenario 3. Noise level: 20%.

3.2 Multitask learning and pre-trained Large Visual Language Model (LVLM)

Current research often focuses on improving the performance of single-task captioning models and overlooks the potential commonalities with other tasks. An example is visual question-answering (VQA), in which a model is trained to answer natural language questions about the content of an image. Such additional tasks, closely related to image captioning, can provide complementary information, thus potentially improving the model's capability and learning speed when trained in a multi-task setting [98]. This phenomenon can be intuitively explained by the model learning hypotheses that explain more than one output, preventing specific tasks from over-fitting [99]. Multitask learning can be especially beneficial in the RS field, where it can alleviate the scarcity of data by aggregating labeled data from different task-specific datasets. Furthermore, multi-tasking can benefit efficiency, as a user may, for instance, desire both a natural language description of an RS image and information extracted through VQA, all from a single model. However, handling multiple tasks with one model poses its own challenges.

Seminal works on multitask learning [100][101] involved using a shared backbone to extract shared features and dedicated heads to learn task-specific outputs. This poses challenges such as the higher complexity of designing dedicated heads, as well as the integration of different loss functions and their balancing for the proper functioning of the multi-task setting. A more recent and direct approach involves casting each downstream task as an auto-regressive sequence prediction problem, which can be solved within a unified sequence-to-sequence model. This approach removes the burden of crafting custom task-specific heads or loss functions, providing a more general framework for multitasking. Image captioning is a natural example of a vision-language task that can be cast as a sequence-to-sequence prediction problem. Others are visual question-answering (VQA) and visual question generation (VQG). Surprisingly, other tasks, such as object detection [102][103], have been analyzed in this context, providing a promising direction for future universal models. A key difference between previous multitask models and this new approach is that a language model can be pointed to solve specific tasks by "tuning" it using *instructions*. As we saw in the introduction, instructions are textual directives that explicitly direct a model to solve a specific task. For example, a natural instruction for image captioning can be "Describe this image in detail." The model learns to answer different queries differently, effectively solving different tasks within the same input-output format. For object detection, an example of an instruction can be "Localize all the swimming pools in this image," to which the model answers with a sequence of coordinates of bounding boxes. This approach has gained interest after the introduction of the so-called foundational models, which are usually very large models trained on vast quantities of data in an unsupervised or semi-supervised way to gather general knowledge. These task-agnostic models, when fine-tuned on high-quality samples of instructions and desired answers, become very robust and powerful *assistants* that can help a user solve many different tasks at the same time just by changing how we *prompt* them. In this chapter, we want to study the feasibility and impact of instruction-tuning a pre-trained large vision-language model on two tasks simultaneously, namely image captioning and VQA. We aim to investigate the impact of jointly addressing both tasks in a unified setting on each other's downstream performance. We also want to assess whether leveraging a large pre-trained model offers advantages over specialized models trained from scratch. We thus adopted LLaVA [104], an open-source large vision-language model that excels in instruction-following image understanding tasks. The authors trained the model in two phases. First, they connected a frozen image encoder (CLIP [91]) to a frozen language model (LLama [105]) and pre-trained a small feed-forward network to adapt the visual tokens extracted from the vision model to the text tokens latent space in the language model. This has been achieved using a $558K$ images subset of the LAION-CC-SBU dataset with synthetic captions from Blip [106]. The goal is to have the language model predict the image caption, relying on the adapted visual tokens, thus aligning the two modalities. After pre-training, the model underwent fine-tuning using a curated dataset of image-related instructions for different tasks such as captioning, visual question answering, complex reasoning,

and detailed description, achieving SoTA performance on several natural image benchmarks. However, despite its impressive capabilities in the natural domain, its performance tends to be suboptimal when applied to RS scenes. This performance gap stems from fundamental differences between RS images and natural images, which can be attributed to the high resolution, diverse scales, and unique acquisition angles of RS images. To bridge this gap and analyze if multitasking can benefit single-task performance, we propose Remote Sensing Large Language and Vision Assistant (RS-LLaVA), a large vision language model tailored for RS image analysis. RS-LLaVA accepts an RS image and a textual instruction as inputs and jointly performs image captioning *or* VQA. We adapt the model following the same two-step strategy as the authors of LLaVA. In the pre-training step, the adapter layer connecting the image encoder and the language decoder is pre-trained on captions from the UCM dataset. Subsequently, we construct an RS instruction-oriented dataset using existing RS image captioning and VQA datasets. Due to the size of the language model (7B or 13B parameters), we fine-tune the model using LoRA [107] to reduce memory requirements. Experimental results demonstrate that RS-LLaVA outperforms previous state-of-the-art methods in single-task and multi-task scenarios.

3.2.1 Instruction dataset

As discussed in the introduction, we need a dataset of *instructions* to solve different tasks using a single vision language model. This is not vastly different from common deep learning training datasets, except that in the case of language models and our case of vision-language models, both the input query and the desired output are treated as sequences of *text* tokens. Examples of instructions to elicit a model to solve the captioning task can be:

- What does this image represent?
- What is the topic of this image?
- Provide a description of the image.

Similarly, examples of queries for VQA can be:

- Is there a ship in the image?
- How many buildings are there?
- What is located in the top-right corner of the image?

Our RS-instructions dataset is constructed by mixing four captioning datasets: two for captioning (UCM-Captions and UAV-Captions) and two for VQA (RSVQA-LR and RSIVQA-DOTA). Table 3.17 provides a summary of the dataset statistics.

UAV-Captions [108] is a remote sensing image captioning dataset captured near the city of Civezzano, Italy, on 17 October 2012, using an unmanned aerial vehicle equipped with an EOS 550D camera. The dataset consists of ten RGB images, each with a resolution of 5184×3456 pixels and a spatial resolution of 2 cm. Out of these ten images, six are designated for training, one for validation, and three for testing. From the original images, crops of size 256×256 pixels are extracted. Specifically, the training images yield a total of 1746 crops, while the testing images provide 882 crops. Each crop is associated with three descriptions written by different annotators.

RSVQA-LR [109] is a VQA dataset consisting of 772 low-resolution images derived from seven tiles captured by the Sentinel-2 satellite, covering an area of 6.55 km^2 over the Netherlands. The images are RGB of size 256×256 pixels, with a resolution of 10 m. The images are split into 572, 100, and 100 for train, validation, and test, respectively. The dataset comprises a total of 77232 questions, with each image annotated with approximately 100 questions. The questions in the dataset cover four categories: object

presence (answer: yes/no), comparisons between objects (answer: yes/no), rural/urban classification (answer: rural/urban), and object counting.

RSIVQA-DOTA [110] is a VQA dataset based on the DOTA [111] object detection dataset. It includes questions about scenes, objects, relative locations, color, and shape. The total number of image/question/answer triplets in the dataset is 16430. The questions are of three types: presence, counting, and other. The dataset is split into three sets: 80% for the train set, 10% for the test set, and 10% for the validation set.

Combining the original datasets' train and test splits, our instruction-based dataset comprises 7058 samples, with 5506 used for training/validation and 1552 for testing.

Table 3.17: Datasets used to build the RS-instructions dataset.

Dataset	**Task**	**#Images**	**Image Size**	**Texts per Image**	**Resolution (m)**
UCM-Captions [35]	Captioning	2100	256 × 256	5	0.3048
UAV-Captions [108]	Captioning	2628	256 × 256	3	0.02
RSVQA-LR [109]	VQA	772	256 × 256	100–101	10
RSIVQA-DOTA [110]	VQA	1868	Varies	3–24	Varies
RS-instructions dataset	Captioning + VQA	7058	Varies	Varies	Varies

(a)

Human: *<image> Is it a rural or an urban area?*
GPT: rural
Human: *How many commercial buildings are there?*
GPT: 8
Human: *Is there a grass area?*
GPT: yes
Human: *What is the amount of water areas?*
GPT: 10

(b)

Human: *<image> What does this image represent?*
GPT: an intersection with a few cars on the road
Human: *What is the topic of this image?*
GPT: an intersection with two roads vertical to each other
Human: *What is the topic of this image?*
GPT: an intersection with lots of cars parked at the corners
Human: *Provide a description of the image*
GPT: an intersection with two roads vertical to each other

Figure 3.11: Samples from the RS-instructions Dataset: (a) Image from RSVQA-LR dataset and (b) Image from UCM Captions dataset. `<image>` is a placeholder to accommodate visual tokens from the encoder at the beginning of the textual query.

A set of template instructions is employed to transform image captioning samples into the instruction-based format. For example, prompts such as "Describe the image" or "What does this image represent?" are used, where the expected response corresponds to the original image caption. For VQA datasets, question-answer pairs are treated as single instruction-response samples. Some examples are provided in Figure 3.11. Our dataset can be accessed at this link.

3.2.2 Methodology

Model Architecture

As depicted in Figure 3.12, RS-LLaVA comprises two modules: a pre-trained visual encoder and a pre-trained large language model decoder. A projection network maps the visual embeddings from the encoder, which are inserted at the start of the stream of textual embeddings to condition the LLM response. The network expects training samples of the form $\{\mathbf{X}, \mathbf{I}, \mathbf{R}\}$, where $\mathbf{X}$ represents the image,

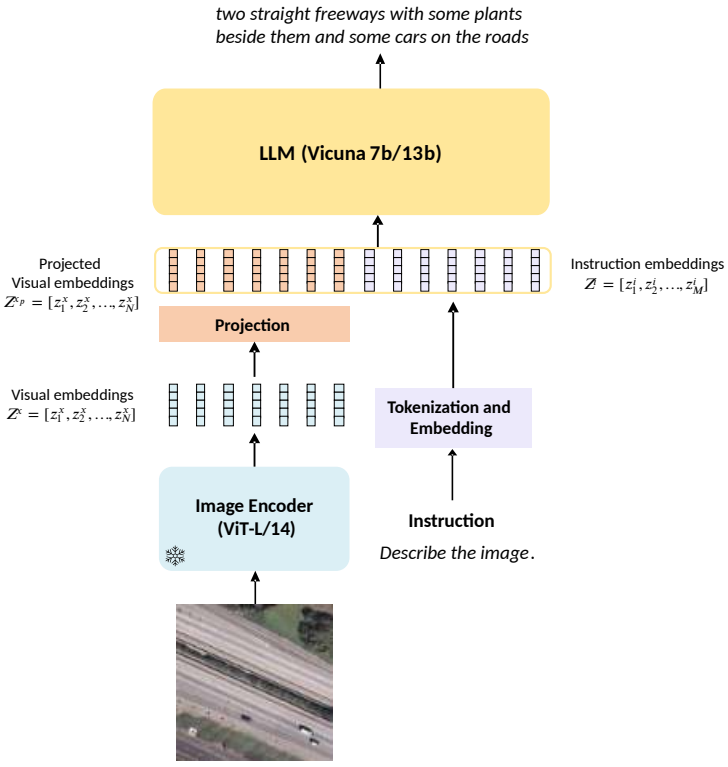


Figure 3.12: RS-LLaVA architecture. CLIP's ViT-L/14 is used as the image encoder, while Vicuna (7b or 13b) is chosen as the decoder. A feed-forward projection layer adapts the visual embeddings from ViT to the text embeddings space of the LLM.

$\mathbf{I}$ represents the query instruction, and $\mathbf{R}$ represents the desired response. The encoder maps the input image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ to a sequence of visual embeddings $\mathrm{Z}^x \in \mathbb{R}^{N \times d_x}$. H, W, and C represent the height, the width, and the number of channels of the image, N is the length of the sequence of visual embeddings, and d_x is their dimensionality. The projection layer, a two-layer feed-forward network, maps the visual embeddings Z^x to the text embedding dimension d_t, creating the sequence $\mathrm{Z}^{x_p} \in \mathbb{R}^{N \times d_t}$. The projected visual embeddings are concatenated with the text embeddings of the instruction $\mathrm{Z}^i \in \mathbb{R}^{M \times d_t}$,

creating the input to the network $\mathrm{Z} = \left[\mathrm{Z}^{x_p}; \mathrm{Z}^{i}\right] \in \mathbb{R}^{(M+N)\times d_t}$. The network is trained to maximize the likelihood of auto-regressively generating the sequence of K tokens of the response $\mathrm{R} = [r_1, r_2, \ldots, r_K]$, as described in Equation 2.2.

Model Training

Following the original training recipe of [104], training involves two steps: 1) pre-training and 2) fine-tuning. During pre-training, the weights of the image encoder and the LLM remain fixed, while the projection network is adapted to the RS data distribution using images and captions from the UCM-Captions dataset. In this step, the goal is to predict the caption of the input image directly without any instruction. In the next step, the projection network and the image encoder are frozen while the LLM is fine-tuned. Given the high number of decoder parameters, we fine-tuned the LLM using the Low-Rank Adaptation (LoRA) technique [107]. LoRA makes it possible to fine-tune large models, limiting memory utilization. This approach works especially well for transformer-based models, which are characterized by large weight matrices in the attention layers. Instead of modifying all parameters, LoRA approximates a large matrix by introducing two low-rank matrices and fine-tuning only these components for the specific task. During inference, the model is "adapted" by fusing the two low-rank matrices with the original pre-trained matrix, effectively acting as a modulation on the original language model.

Formally, given a pre-trained weight matrix $W_0 \in \mathbb{R}^{U\times V}$, two low-rank matrices B and A acts as a modulation on the fixed pre-trained matrix:

$$\widetilde{W_0} = W_0 + \Delta W = W_0 + BA, \tag{3.8}$$

with $B \in \mathbb{R}^{U\times R}$, $A \in \mathbb{R}^{R\times V}$, and the rank $R \ll \min(U, V)$. We keep the default LoRA configuration for transformer-based large language models, applying LoRA on query, key, and value projection matrices in each attention layer: W_q, W_k, and W_v.

3.2.3 Experimental Results

Setting

For image encoding, we adopted the pre-trained vision backbone of CLIP-ViT-L/14 [91], with an image resolution of 336×336. This backbone divides the image in a grid of 14×14 pixels, resulting in a visual embedding sequence length of $N = 576$. For the language model, we tested the 7B and 13B parameter variants of Vicuna-v1.5 [112]. The visual embedding dimension is $d_x = 768$, while the text embedding dimension is $d_t = 5120$. We employ LoRA with rank $r = 64$ and $\alpha = 16$, as suggested in the original paper [107]. We trained for 15 epochs in each phase (pre-training and fine-tuning), using the Adam optimizer, with a learning rate of $1e-4$.

Evaluation metrics

We utilize different metrics to evaluate the performance of our model in the captioning and VQA tasks. We assess the captioning performance using the following standard metrics: BLEU 4, METEOR, ROUGE-L, and CIDEr. For the VQA task on VQA-LR, we use accuracy for each question type. On VQA-DOTA, for yes/no questions, we use precision, recall, and F1 score, while on counting questions, we use the root mean square error (RMSE) with the ground truth answer.

For every experiment, we report the performance of different decoder sizes (7B and 13B). The analysis differentiates between two scenarios: (1) when the model is trained on the constructed RS-instructions dataset to perform multiple tasks (Joint), and (2) when the model is trained on a single task (Single).

Image Captioning Results

Table 3.18: Captioning results on UCM-Captions. Bold highlights the best results.

Training	Decoder	Bleu 4	METEOR	ROUGE-L	CIDEr	Training Time
Single	Vicuna7B	72.78	46.75	83.72	343.21	2.10 h
	Vicuna13B	74.50	48.62	**86.09**	355.06	3.73 h
Joint	Vicuna7B	72.84	47.98	85.17	349.43	11.04 h
	Vicuna13B	**76.03**	**49.21**	85.78	**355.61**	19.40 h

Table 3.19: Captioning results on UAV-Captions. Bold highlights the best results.

Training	Decoder	Bleu 4	METEOR	ROUGE-L	CIDEr	Training Time
Single	Vicuna7B	**53.27**	42.03	78.21	**427.17**	1.83 h
	Vicuna13B	52.79	**42.77**	**79.15**	423.18	3.81 h
Joint	Vicuna7B	49.02	40.46	76.80	404.54	11.04 h
	Vicuna13B	49.24	40.14	76.28	390.30	19.40 h

Results on image captioning, depicted in Tables 3.18 and 3.19, show a mixed trend. On UCM-Captions, integrating the two tasks led to better performance, especially for the $13B$ parameters decoder. This is, however, in contradiction with the results on UAV-Captions, where it seems that integrating the two tasks led to a worse performance. This dichotomy can be explained by the affinity of the two image captioning datasets with the VQA datasets. Indeed, UCM Captions images are more similar to the ones in the VQA-DOTA dataset, which can result in a better sharing of complementary information between these two datasets. In contrast, images from the UAV dataset show differences in viewpoint and resolution with the other datasets, which can explain the reduced performance due to worse information sharing. Furthermore, it is evident how, for UCM-Captions, the size of the decoder is directly proportional to the performance, while on UAV-Captions, this trend is much less pronounced. As expected, the joint training on both tasks leads to a longer training time, and the $13B$ variant is much more computationally demanding than the $7B$ variant. All these observations should be kept in place when choosing the right model to fine-tune, especially considering the amount and type of data at our disposal. Qualitative results of our model on the image captioning task are reported in Figure 3.13.

VQA Results

Table 3.20: VQA results on RSIVQA-LR. Avg: average of the accuracies on different question types. Bold highlights the best results.

Training	Decoder	Count	Pres.	Comp.	Urban/Rural	Avg.	Training Time
Single	Vicuna7B	75.05	**92.97**	91.23	**95.00**	88.56	3.25 h
	Vicuna13B	**75.87**	92.32	**91.37**	**95.00**	**88.64**	7.10 h
Joint	Vicuna7B	74.38	92.80	91.33	94.00	88.13	11.04 h
	Vicuna13B	73.76	92.27	**91.37**	**95.00**	88.10	19.40 h

Table 3.21: VQA results on RSIVQA-DOTA. Bold highlights the best results.

Training	Decoder	Count	Yes/No			Training time
		RMSE	P	R	F1	
Single	Vicuna7B	**221.40**	91.49	72.07	80.63	1.95 h
	Vicuna13B	226.79	89.03	**82.79**	**85.80**	3.88 h
Joint	Vicuna7B	**209.47**	85.26	**86.15**	85.70	11.04 h
	Vicuna13B	232.75	**100**	33.28	49.94	19.40 h

In this experiment, we evaluate the performance of our model in answering natural language questions about remote sensing (RS) images. The results are presented in Tables 3.20 and 3.21.

On the RSVQA-LR dataset, joint fine-tuning yields worse results than fine-tuning on the single dataset. These findings suggest that, for RSVQA-LR, the additional information provided by the other datasets does not significantly enhance VQA performance. In this case, images from RSVQA-LR are different from the other datasets since the resolution is much coarser. Similarly to the case of image captioning on UAV-Captions, this difference can result in worse sharing of information between the datasets, explaining the performance drop. Notably, the performance difference between the two model sizes is minimal in both training scenarios.

In contrast, on the RSIVQA-DOTA dataset, training on the joint dataset improves the performance of the $7B$ variant, which achieves an F1 score comparable to that of the $13B$ variant trained on the single task while also delivering a much lower RMSE on counting questions. This supports our hypothesis that the stronger compatibility between RSIVQA-DOTA images and UCM-Captions facilitates effective knowledge transfer between the datasets, thereby enhancing single-task performance. This behavior contrasts with RSVQA-LR, whose images are of lower spatial resolution than those in the other datasets, potentially limiting the benefits of cross-task information sharing. Some qualitative results on both RSVQA-DOTA and RSVQA-LR are shown in Figure 3.14.

3.2.4 Comparison with state of the art

In this section, we compare the results of our RS-LLaVA model trained on our RS instruction dataset with other SoTA models in RS image captioning and VQA. Captioning results are reported in Tables 3.22 and 3.23. On UCM-Captions, our $13B$ model outperforms several strong baselines on all metrics except for CIDEr, where it achieves the second position. Both variants perform satisfactorily, with the $7B$ variant very close to the second-best algorithm [52]. On UAV-Captions, both variants of our model surpass the previous state-of-the-art (SoTA) results by a large margin. VQA results on the RSVQA-LR dataset are reported in Table 3.24. The results show that both variants of our model offer more accurate answers compared to other state-of-the-art methods across all question types.

3.2.5 Conclusions

This chapter explored the promising capabilities of LLMs and their extension, LVLMs, in the field of RS, specifically by investigating their multi-tasking potential for tasks like image captioning and VQA. We introduced RS-LLaVA, an enhanced version of LLaVA adapted for RS imagery. To train this model, we developed the RS-instructions dataset by leveraging existing four single-task datasets. We have demonstrated the capability of the proposed architecture using two different variants of increasing size, namely vicuna-7B and vicuna-13B. While the experiments demonstrated the notable performance of the proposed RS-LLaVA architecture, it is important to mention the computational challenges posed by large

Table 3.22: Results of different RS image captioning methods on the UCM-Captions dataset. The best results are highlighted in bold, while the second-best results are underlined.

Method	**BLEU4**	**METEOR**	**ROUGE**	**CIDEr**
CSMLF [113]	12.10	13.20	39.27	22.27
VGG19+LSTM [35]	21.90	20.60	-	45.10
GoogleNet-hard att. [36]	65.62	44.89	79.62	320.01
VAA [114]	63.87	43.80	78.24	339.46
Yuan *et al.* [115]	66.23	43.71	77.63	316.84
ResNet18 MSF [116]	63.45	-	73.18	329.56
SD-RISC [117]	53.80	39.00	69.50	213.20
Hoxha *et al.* [108]	37.02	37.02	67.87	292.28
Li *et al.* [118]	67.00	47.75	75.67	285.47
MSA [119]	70.21	45.04	79.18	325.71
Word-sentence [120]	62.02	43.95	71.32	278.71
Structured att. [121]	71.49	46.32	81.41	334.89
Zia *et al.* [45]	67.50	44.60	-	323.10
Li *et al.* [122]	69.76	45.71	80.72	338.87
SCST [123]	67.60	-	76.00	336.00
Wang *et al.* [124]	65.10	45.30	78.50	338.10
MLCA [41]	66.80	43.50	77.20	324.00
Ye *et al.* [52]	<u>73.76</u>	<u>49.06</u>	83.64	**371.02**
RSLLaVA-7B (Joint)	72.84	47.98	<u>85.17</u>	349.43
RSLLaVA-13B (Joint)	**76.03**	**49.21**	**85.78**	<u>355.61</u>

Table 3.23: Results of different RS image captioning methods on the UAV dataset. The best results are highlighted in bold, while the second-best results are underlined.

Method	**BLEU4**	**METEOR**	**ROUGE**	**CIDEr**
Hoxha *et al.* [108]	39.22	32.81	69.63	391.31
Hoxha *et al.* [108]	39.69	32.17	69.31	389.45
Basmal *et al.* [P4]	45.17	38.18	75.19	390.27
RSLLaVA-7B (Joint)	<u>49.02</u>	**40.46**	**76.80**	**404.54**
RSLLaVA-13B (Joint)	**49.24**	<u>40.14</u>	<u>76.28</u>	<u>390.30</u>

Table 3.24: Results of different VQA models on the RSVQA-LR dataset. The best results are highlighted in bold, while the second-best results are underlined.

Method	**Count**	**Presence**	**Comparison**	**Urban/Rural**	**Average**
Lobry *et al.* [125]	67.01	87.46	81.50	90.00	81.49
Yuan *et al.* [126]	68.53	90.13	86.91	92.00	84.39
Bazi *et al.* [127]	72.22	91.06	91.16	92.66	86.78
RSLLaVA-13B (Joint)	**74.38**	**92.80**	<u>91.33</u>	<u>94.00</u>	**88.13**
RSLLaVA-13B (Joint)	<u>73.76</u>	<u>92.27</u>	**91.37**	**95.00**	<u>88.10</u>

Instruction: Provide a description of the image.
Response: There are many airplanes stopped at the airport.
Ground Truth: There are many airplanes at the airport.

Instruction: What does this image represent?
Response: There are two straight freeways in the desert.
Ground Truth: There are two straight freeways in the desert.

Instruction: Provide a description of the image.
Response: Some asphalt on the left and some trees on the right.
Ground Truth: Small road on the left and several trees on the right.

Instruction: What does this image represent?
Response: Soil field with building facade on the top.
Ground Truth: Soil ground with building facade on the top.

Figure 3.13: Sample of RS-LLaVA captioning results from UCM-Captions (top row) and UAV-Captions (bottom row).

parameter sizes. Indeed, LLMs often require extensive computational resources for training and inference, limiting their accessibility and scalability. To address this issue in future research, efforts should focus on exploring techniques for model compression, such as knowledge distillation or parameter pruning, to reduce the computational burden while maintaining performance. Additionally, one can plan to integrate additional datasets and tasks, such as visual grounding and change detection in multi-temporal images, to further enhance the versatility and applicability of RS-LLaVA in RS applications.

	
Instruction: How many small vehicles are there in this picture? **Response:** 20 **Ground Truth:** 2	**Instruction:** How many planes are there in this picture? **Response:** 3 **Ground Truth:** 3
Instruction: How many large vehicles are there in this picture? **Response:** 1 **Ground Truth:** 16	**Instruction:** How many small vehicles are there in this picture? **Response:** 8 **Ground Truth:** 2
Instruction: How many roundabouts are there in this picture? **Response:** 2 **Ground Truth:** 1	**Instruction** How many large vehicles are there in this picture? **Response:** 3 **Ground Truth:** 2
Instruction: Does this picture contain small vehicle? **Response:** Yes **Ground Truth:** Yes	**Instruction:** How many helicopters are there in this picture? **Response:** 3 **Ground Truth:** 2
	
Instruction: What is the amount of large water areas? **Response:** 6 **Ground Truth:** There are 6 large water areas	**Instruction:** Is it a rural or urban area? **Response:** Urban **Ground Truth:** It is an urban area
Instruction: Are there more water areas than roads? **Response:** No **Ground Truth:** No there are not more water areas than roads	**Instruction:** Is a commercial building present? **Response:** Yes **Ground Truth:** Yes a commercial building is present
Instruction: Is there a large road? **Response:** No **Ground Truth:** No there is no large road	**Instruction** Is a residential building present? **Response:** Yes **Ground Truth:** Yes a residential building is present
Instruction: What is the number of large forests? **Response:** 7 **Ground Truth:** There are 6 large forests	**Instruction:** What is the amount of grass areas? **Response:** 326 **Ground Truth:** There are 268 grass areas

Figure 3.14: RS-LLaVa VQA results from RSVQA-DOTA (top row) and RSVQA-LR (bottom row).

Chapter 4

Enrich captions with Visual Dialogue

[P5] Riccardo Ricci, Yakoub Bazi, and Farid Melgani. "Machine-to-machine visual dialoguing with ChatGPT for enriched textual image description". In: Remote Sensing 16.3 (2024), p. 441.

4.1 Introduction

While accuracy and robustness are very important for image captioning, another limitation of RS image captioning is the restrictiveness of the ground truth captions with respect to the richness of the visual scene. Most datasets from image captioning literature have captions composed of a single sentence describing the general content and appearance of the main objects in the scene. Oftentimes, when the main concepts are similar, the caption cannot capture details that differentiate one image from the other. Figure 4.1 depicts some examples from UCM-Captions. These images share most ground truth captions, but their content is rarely equivalent. For example, the first row depicts different crop types, with some being low plants and some trees, but the captions do not grasp these differences. In the second row, a pool is visible in the first image, which is overlooked by the captions, as well as the layout of the residential area. A direct drawback of this restrictiveness in the ground-truth captions is the reduced information conveyed to the user. On the other hand, this can also impact other applications such as text-to-image retrieval, representation learning, and so on. The descriptiveness of captions is inherently influenced by the data on which the model has been trained. To address the limited caption descriptiveness, some studies are introducing new datasets featuring longer and more detailed captions. An example is [128], in which the authors propose a labeled dataset of 2585 images with long and detailed captions. However, this approach presents one main limitation: curating such a dataset is extremely time-consuming. Visual dialogue [129] has been recently introduced as a paradigm in which a machine seamlessly interacts with human users using natural language, holding a conversation about an image's contents. Specifically, given an image, a dialog history, and a follow-up question about the image, the task is to answer a follow-up question. Nowadays, machines trained to solve such a task are called visual assistants. In recent years, tremendous progress has been made in conversational models, both text-only [74][105] and visual assistants [130][104][86]. For example, authors in [74] used supervised learning and reinforcement learning from human feedback to teach an LLM to mimic the human ability to engage in dialogues, comprehending and answering diverse queries formulated in natural language. Authors of [104] collected a large-scale dataset of synthetic conversations about visual content using GPT-4 [130], then used this data to teach an LLM to hold conversations about image contents and answer a diverse set of questions regarding the image content. In [86], the authors connect a frozen LLM to a frozen image encoder, enabling Blip-2 to solve diverse zero-shot queries about the image content, thanks to the ability of the LLM to follow instructions. Thanks to the remarkable ability of those models to follow instructions,

Figure 4.1: Images with ground truth captions. The other three images share most of the captions despite being different from the first image.

in this chapter, we envision a visual dialogue system entirely based on such machines. The idea is to leverage a text-only LLM to generate follow-up questions and a visual assistant to answer those questions based on the image. A visual dialogue is carried out to explore further information besides that included in the starting "plain" caption. This is similar to how we as humans interact with each other to acquire information about the world that surrounds us: we ask questions, and we receive answers. This way, we try to reach the same goal of richer and more comprehensive captions, removing the burden of manually crafting a targeted dataset.

We argue that dialogue can be beneficial for two reasons: (1) it decomposes the problem into sub-problems represented by various questions, whose solutions can be easier, and (2) it enables exploring to a deeper degree the semantic content hidden in an image. In this chapter, we explore different solutions

for enriched remote sensing image description under the paradigm of "description through dialogue," providing strengths and weaknesses of each, as well as directions for further research.

4.2 Methodology

The general idea, as depicted in Figure 4.2, is to establish a machine-to-machine (M2M) visual dialogue (VD) that requires no (or little) human intervention and can extract additional information to enrich an initial "plain" caption. The initial "plain" caption sparks the dialogue, which serves to further explore the image contents. The information collected in the exchange is ultimately summarized to generate the final output. Within the visual dialogue framework, exploration is performed in steps; each question-answer pair explores additional concepts and attributes, possibly building on previously extracted knowledge to decide which follow-up questions to ask. In this chapter, we propose three different strategies to establish a dialogue about the contents of an image: Open-Ended Dialogue (OED), Closed-Form Dialogue (CFD), and Closed-Form Dialogue with Context (CFD-C). In Open-Ended Dialogue, similarly to [131], we envision a system comprised of two modules, one devoted to asking questions and the other that, based on a given question, provides an appropriate answer (conditioned on the target image). This conceptual

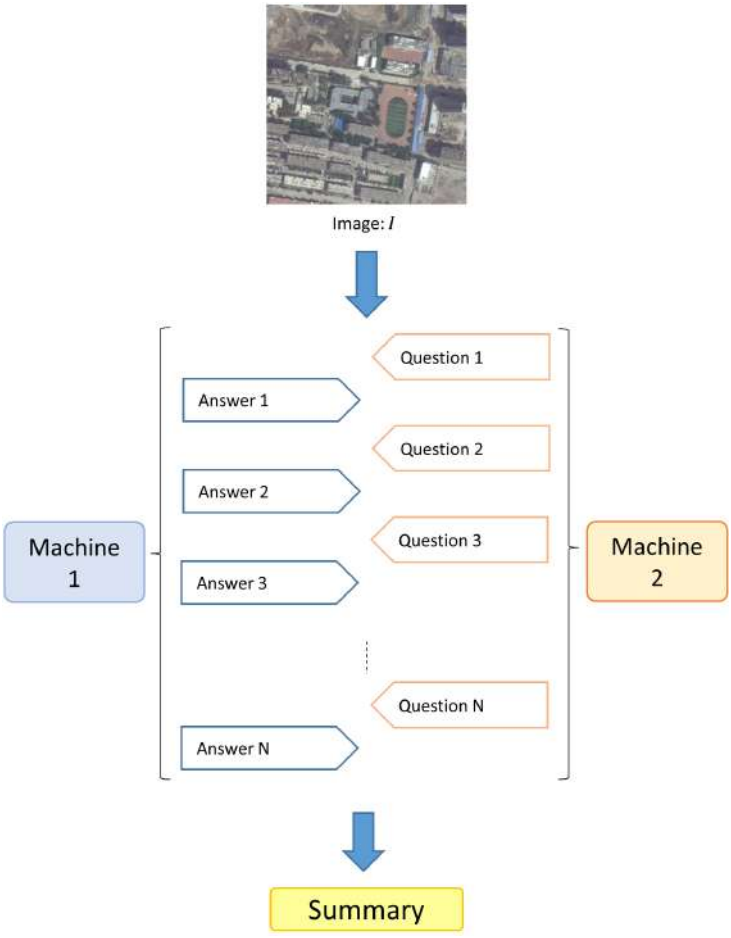


Figure 4.2: Conceptual representation of the machine-to-machine visual dialoguing (M2M-VD) paradigm.

division keeps the individual responsibilities of the questioner and answerer isolated. However, this is not a prerequisite: the two roles can also be accomplished within a unified model that provides questions and answers (for example, using a single visual assistant). In Closed-Form Dialogue, predefined template questions are used on each target image. In Closed-Form Dialogue with Context, we try to integrate the context given by the previous questions and answers in the answering process. We explore the strengths and drawbacks of the proposed strategies, trying to set the path for more exploration of this promising but challenging paradigm.

Models used

In our proposals, ChatGPT [74] takes the role of the questioner, while Blip-2 [86] is adopted as the answerer.

ChatGPT [74] is a fine-tuned version of GPT-3.5 [72], and it belongs to the family of text-only instruction-tuned assistants. GPT-3.5 is a large language model pre-trained to perform the next token prediction task on a large dataset of web-scraped textual data. Starting from the pre-trained GPT-3.5 weights, authors of [74] use supervised learning and reinforcement learning from human feedback (RLHF) to confer instruction-following capabilities to the LLM.

Blip-2 is a large vision language model that bridges visual and text modalities by connecting a frozen image encoder to a frozen large language model (LLM). The goal is to preserve the reasoning ability of large language models while injecting image understanding by adaptively merging the two modalities through the so-called "query transformer." The alignment between the image encoder and the LLM follows a two-step procedure, using a corpus of 129 million of images with corresponding captions. In the first step, the visual features extracted by a frozen image encoder are forwarded to the query transformer, which modifies them to suit three pre-training tasks: (1) image-text contrastive learning (ITC), (2) image-grounded text generation (ITG), and (3) image-text matching (ITM). ITC aims to enhance the mutual information between image and text representations of corresponding (positive) pairs while reducing that of non-corresponding (negative) pairs. ITG trains the query transformer to generate texts conditioned on the input image. This objective forces the queries to capture the most meaningful clues to generate the ground truth description from the frozen image encoder representation. ITM consists of a binary classification of whether the image corresponds to the text (positive pair) or not (negative pair). The prediction is achieved by generating scores for each query vector in the Query Transformer and averaging to obtain the overall matching result. In the second step, the authors connect the query transformer to a frozen large language model to align the large language model in generating text conditioned on the queries extracted by the query transformer. According to the authors, keeping the models frozen during alignment can mitigate the catastrophic forgetting problem, preserving the ability to perform prompt-based text generation while including conditioning information from the image.

Open-Ended Dialogue (OED)

Figure 4.3 depicts the scheme of OED. In this framework, the question generation process is dynamically controlled by ChatGPT, which relies on the dialogue history (questions and answers). Specifically, the following steps are performed:

1. A fixed prompt, "Describe this image in detail," is used to spark the conversation.
2. Blip-2 answers with the first "plain" caption.
3. ChatGPT, using the "plain" caption as context, generates a follow-up question to request additional details.
4. Blip-2 answers by looking at the image.

5. ChatGPT, using the "plain" caption and the dialogue history as context, produces a follow-up question to further explore the image contents.

6. Blip-2 answers by looking at the image.

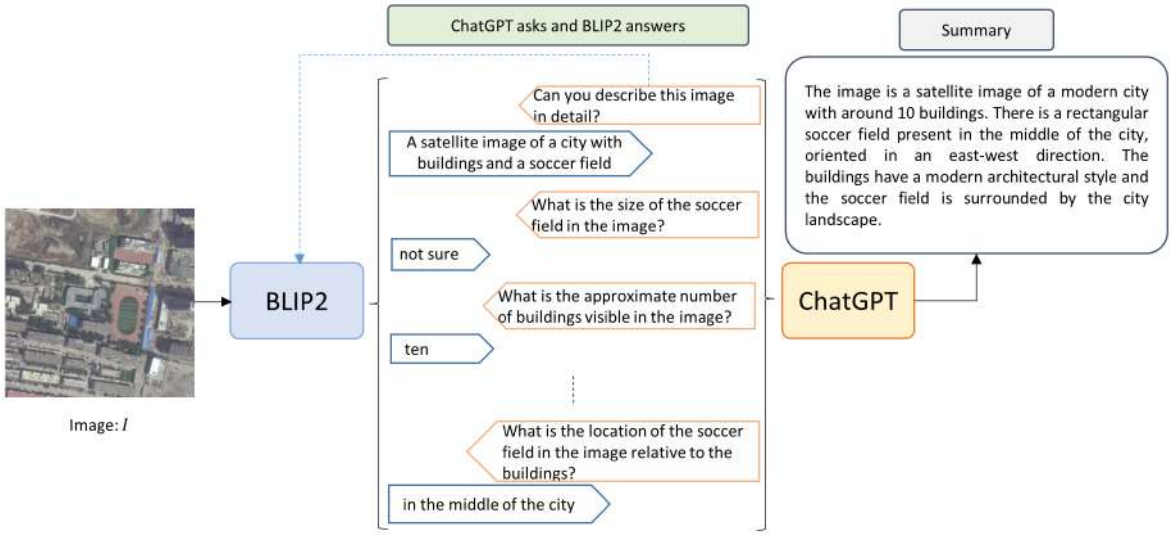


Figure 4.3: Open-ended dialogue block scheme.

Steps (5) and (6) are iteratively repeated $K = 10$ times to simulate a visual dialogue. In this preliminary study, the number of iterations is fixed to avoid the complexity of implementing a more advanced stopping criterion. At the end of the dialogue, the entire conversation is concatenated, and ChatGPT is employed to summarize the content using an appropriate prompt to generate the final description.

4.2.1 Closed-Form Dialogue (CFD)

In the Open-Ended Dialogue approach, a significant limitation arises from the complete reliance on the initial description when generating follow-up questions. If the initial description is inaccurate, includes hallucinations, or introduces incorrect concepts, the dialogue may diverge entirely from the actual content of the image. To address this, we propose an alternative approach - Closed-form dialogue - which decouples question generation from the dialogue history. Specifically, the user defines a fixed set of questions to be applied uniformly across all images. CFD can be particularly advantageous in scenarios where (1) the user requires specific information about the scenes or (2) the initial description generated by Blip-2 is insufficient or inaccurate, hindering the generation of meaningful follow-up questions. In general, if the predefined set of questions is well-designed for the target images, this approach facilitates a more consistent extraction of information, thereby improving the reliability of the results.

The process is illustrated in Figure 4.4. Blip-2 is used to answer each predefined question sequentially. As in Open-Ended Dialogue, ChatGPT summarizes the dialogue content and generates the final description.

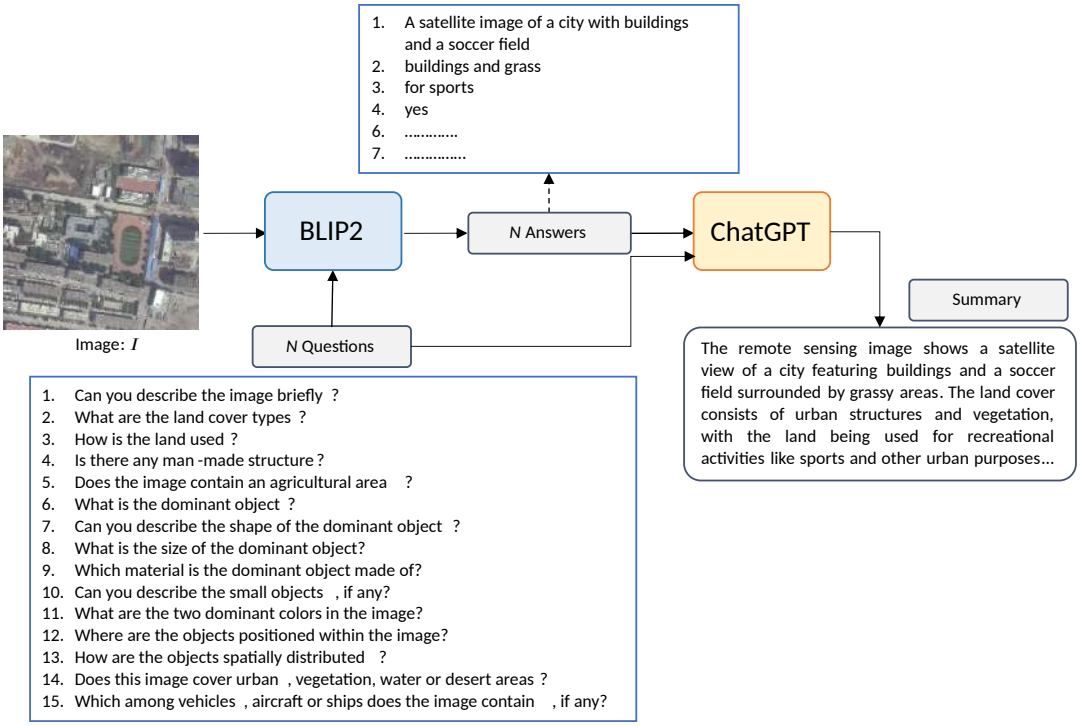


Figure 4.4: Closed-form dialogue block scheme.

4.2.2 Closed-Form Dialogue With Context (CFD-C)

In closed-form Dialogue with Context, we try to move one step forward in exploring the influence of the dialogue history on answers to subsequent questions. The two previous strategies consider the dialogue history only when generating the questions, not answering them. To explore this influence, we built an encoder-decoder architecture, depicted in Figure 4.5, using the ViT-L/14 vision transformer backbone of CLIP [91] as the image encoder and GPT-2 [62] as the language model decoder. The model's structure is very similar to RSLLaVA, as seen in the previous chapter, and thus, the operations are carried out to generate the answers. The vision transformer encodes an image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ into a sequence of visual embeddings $\mathrm{Z}^x \in R^{N \times d_x}$ where N is the sequence length and d_k is the dimensionality of the visual embeddings. Similarly, the first question $\mathbf{Q^1}$ is converted in a sequence of text embeddings $\mathrm{Z}^{q1} \in R^{M \times d_t}$ where M is the sequence length and d_t is the dimensionality of the text embeddings. The visual embeddings are concatenated with the text embeddings of the first question $\mathrm{Z} = [\mathrm{Z}^x, \mathrm{Z}^{q1}]$ and used as input features to GPT-2, which auto-regressively generates the tokens of the first answer $\hat{\mathrm{A}}^1 \in R^P$, where P is the sequence length.

To predict the answer to the second question, the context is obtained by concatenating the embeddings of the image, the first question, the first answer, and the second question: $\mathrm{Z} = [\mathrm{Z}^x, \mathrm{Z}^{q1}, \hat{\mathrm{Z}}^{a1}, \mathrm{Z}^{q2}]$. Having as input Z, GPT-2 is trained to generate the tokens of the second answer $\hat{\mathrm{A}}^2$. The process continues for subsequent answers till the end of the dialogue. As for the previous approaches, we use ChatGPT to summarize the dialogue and generate the final output.

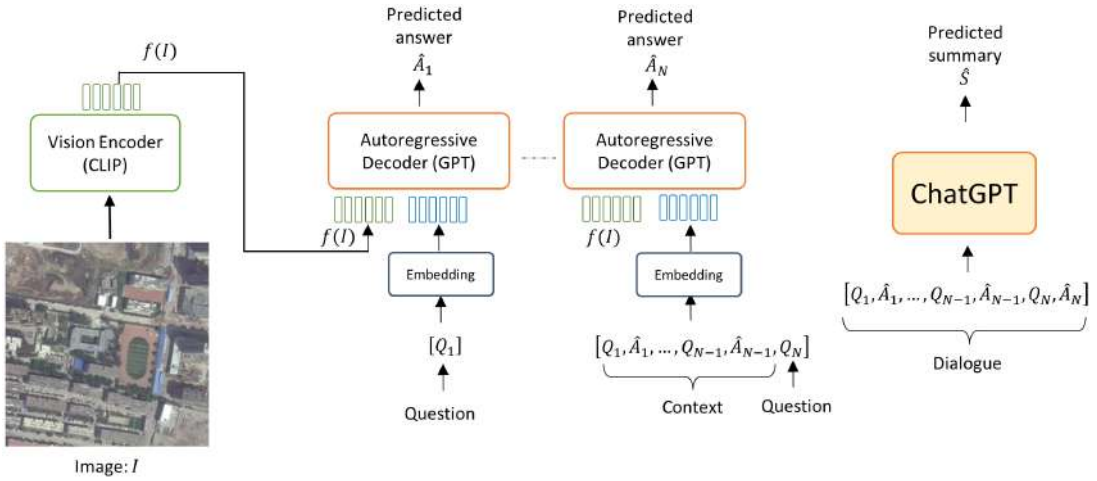


Figure 4.5: Block scheme of closed-form dialogue with context during inference.

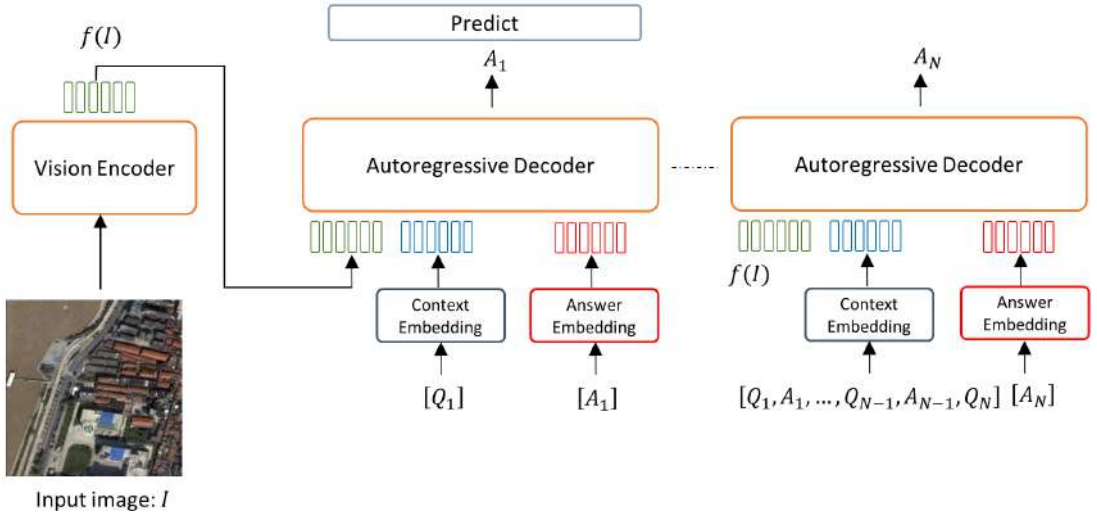


Figure 4.6: Closed-form dialogue with context during training.

4.2.3 Dataset And Metrics

We tested the dialoguing framework on two remote sensing image captioning datasets: RSICD [36] and UCM-Captions [35]. The choice has been made to have two datasets almost an order of magnitude different in size and analyze how this can impact the performance.

As stated in [131] and [132], assessing the performance in dialogue-based image description is a daunting problem, mainly because of the lack of ground-truth summaries to compare with. As explored in [131], using traditional metrics leads to huge performance drops despite the result's coherence. In both these studies, authors use human assessment to evaluate the descriptions and the dialogues, which is expensive and time-consuming. Due to the lack of resources, we were forced to experiment with other metrics. We identified two existing reference-free metrics: CLIPScore [133] and text-to-image (T2I) retrieval as feasible candidates. Furthermore, we propose a novel reference-free metric, which consists of

generating synthetic images from the final textual output and comparing synthetic and original images. In the analysis, we address their strengths and weaknesses in evaluating such results.

CLIPScore

Given an image $\mathbf{X}$ and a text $\mathbf{T}$, CLIPScore leverages a pre-trained CLIP [91] model and measures the compatibility between the image and the text. As in equation 3.1, the text branch of CLIP projects a text $\mathbf{T}$ into a vector representation Z^t_{class}. Similarly, the image is projected into a vector representation Z^x_{class}. CLIPScore is computed as follows:

$$\text{CLIPScore}(Z^x_{class}, Z^t_{class}) = c \cdot \max\left(\text{cosine similarity}\left(Z^x_{class}, Z^t_{class}\right), 0\right) \tag{4.1}$$

Where the cosine similarity is computed as in equation 3.2. The $c = 2.5$ is a constant stretching the measure in the $[0-1]$ range. In their paper, the authors of CLIPScore demonstrate that higher CLIPscores correlate with better image descriptions. To have a single aggregated value for an entire dataset, we take the average for all the images.

Text-to-image (T2I) retrieval

As a second metric, we adopt text-to-image (T2I) retrieval as an indirect measure of description accuracy. Suppose you have an image $\mathbf{X}$ and a generated text describing the image $\mathbf{T}$. If the description $\mathbf{T}$ can isolate the image $\mathbf{X}$ from a dataset of other images $[\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_N}]$, this means that it must be coherent and truthfully descriptive of the image contents. Suppose that the same description is generated for every image; this retrieval operation is purely a matter of chance. The more a description is specific to an image, the more effective it is to isolate the image from the whole dataset. We use CLIP-ViT-L/14 to extract embeddings of images and generated captions and calculate recall scores for different levels of strictness: R@1, R@5, and R@10. R@10 means the target image is in the first 10 ranked images, and so on for the other recall scores.

In detail, consider a dataset of N images, represented as $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, where $\mathbf{X}_i$ denotes the i-th image. Each image $\mathbf{X}_i$ is associated with a generated description $\hat{\mathbf{T}}_i$. Using the vision branch of CLIP, the entire dataset of images $\mathbf{X}$ is embedded into a matrix $\mathbf{Z^x} \in \mathbb{R}^{N \times d_x}$. Each row of this matrix, $\mathrm{Z}^x_{class,j} \in R^{d_x}$, corresponds to the visual embedding representation of the j-th image.
For an image i, the retrieval process is as follows:

1. The generated text description $\hat{\mathbf{T}}_i$ is converted into the respective text embedding $\mathrm{Z}^t_{class,i} \in \mathbb{R}^{d_t}$.

2. The similarity between the text embedding $\mathrm{Z}^t_{class,i}$ and the embedding of each image $\mathrm{Z}^x_{class,j}$, is measured computing the cosine similarity $\cos\_\mathrm{sim}(\mathrm{Z}^t_{class,i}, \mathrm{Z}^x_{class,j})$, between the respective embeddings, as described in Equation 3.2. This yields a similarity vector $\mathrm{S}_i \in \mathbb{R}^N$, where the j-th element $\mathrm{S}_{i,j}$ represents the similarity between $\hat{\mathbf{T}}_i$ and $\mathbf{X}_j$:

$$\mathrm{S}_i = \left[\cos\_\mathrm{sim}(\mathrm{Z}^t_{class,i}, \mathrm{Z}^x_{class,1}), \cos\_\mathrm{sim}(\mathrm{Z}^t_{class,i}, \mathrm{Z}^x_{class,2}), \ldots, \cos\_\mathrm{sim}(\mathrm{Z}^t_{class,i}, \mathrm{Z}^x_{class,N})\right] \tag{4.2}$$

3. Entries of S_i are sorted in descending order to determine the rank of each image based on its similarity to the text $\hat{\mathbf{T}}_i$. Let $\mathrm{rank}_i(j)$ denote the rank of the j-th image for the i-th query. Recall@m measures the percentage of queries for which the correct image $\mathbf{X}_i$ (i.e., the image paired with $\hat{\mathbf{T}}_i$) appears within the top m ranked images. Mathematically, we define Recall@m as:

$$\text{Recall@}m = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left\{\mathrm{rank}_i(i) \leq m\right\} \tag{4.3}$$

Where $\mathbf{1}(\cdot)$ is the indicator function, which equals one if the condition inside holds true and zero otherwise.

Text-to-Image and Comparison (T2IC)

The last measure is a novel proposal. The idea is that if an image can be faithfully reconstructed from its description, the description is a faithful representation of its contents. This new evaluation method for caption quality, which we term "Text-to-Image and Comparison" (T2IC), is based on generating synthetic images from the generated descriptions and comparing them with the original images. The higher the similarity, the more faithful the description captures the image content. We fine-tune a pre-trained Stable Diffusion model [134] for text-to-image generation to adapt it to generate synthetic remote sensing RGB images. We fine-tuned Stable Diffusion using images and ground-truth captions from the union of UCM-Captions and RSICD-Captions. We fine-tuned for 20 epochs using the AdamW optimizer with a $1e^{-5}$ learning rate. After obtaining a suitable model to generate synthetic RS images from textual descriptions, we articulate our measure in two values. The first is the average cosine similarity between the synthetic and the original images. The second is the so-called FID score, widely used to compare image generation models. The FID score measures how closely the distribution of synthetic images matches the distribution of real (original) images. Using a pre-trained InceptionV3 [135] model, it converts images into high-level semantic embeddings, as explained in Chapter 1. Specifically, it keeps the InceptionV3 last layer input representation as the semantic visual embedding on the image. It then fits two multivariate Gaussian distributions on top of the extracted features, calculating their mean vectors and covariance matrices. Then, the FID score measures the distance between these distributions as:

$$\text{FID} = \|\text{M} - \hat{\text{M}}\|_2^2 + \text{Tr}\left(\mathbf{C} + \hat{\mathbf{C}} - 2\left(\mathbf{C}\hat{\mathbf{C}}\right)^{\frac{1}{2}}\right) \tag{4.4}$$

Where M and $\hat{\text{M}}$ are the mean vectors, and $\mathbf{C}$ and $\hat{\mathbf{C}}$ are the covariance matrices of the real and synthetic feature distributions, respectively. A lower FID score indicates that the distribution of the semantic visual embeddings extracted from the synthetic set closely matches the one from the original images, signifying higher-quality and more realistic synthetic images.

4.2.4 Prompts

Large language models (LLMs) have demonstrated remarkable prompt-following capabilities, where prompts—textual instructions provided before a request—guide the model in generating responses aligned with the desired task. Blip-2 and ChatGPT leverage LLMs, making prompts a crucial component of their workflows to effectively address user queries. In [131], the authors employed specialized prompts for question and answer generation. Building on this approach, we developed tailored prompts to elicit the questioner and answerer LLM models in our framework.

(1)	I have an image. Ask me questions about the content of this image. Carefully ask me informative questions to maximize your information about this image content. Each time, ask one question only without giving an answer. Avoid asking yes/no questions. I'll put my answer beginning with 'Answer':
(2)	Next Question. Avoid asking yes/no questions. Question:
(3)	Now summarize the information you get in a few sentences. Ignore the questions with answers no or not sure. Don't add information. Don't miss information. Summary:

Table 4.1: Questioner prompts, applied to ChatGPT in our study.

Prompt (1) is used at the beginning of the dialogue to provide information about the user requirements and the desired format of the questions. Prompt (2) is to gather follow-up questions, and it is placed after the previous dialogue history to ask the model to provide another question in the correct format. Prompt (3) is used at the end of the dialogue to summarize the dialogue history in a descriptive paragraph.

(4)	Answer given questions. If you are not sure about the answer, say you don't know honestly. Don't imagine any contents that are not in the image.

Table 4.2: Answerer prompts, applied to Blip-2 in our study.

Prompt (4) is used to condition Blip-2 to output an answer only if it is sure that it is the correct answer. In [131], the authors show that using these prompts reduces hallucinations from the answerer.

4.3 Results

In Tables 4.3 and 4.4, we present quantitative results for various dialogue-based approaches: open-ended dialogue (OED), closed-form dialogue (CFD), and closed-form dialogue with context (CFD-C). These methods are evaluated against two baselines: the "plain" captions generated by Blip-2 and MLAT. We included MLAT as a baseline due to its specialization in remote sensing scenes, lacking in Blip-2. By incorporating these baselines, we aim to determine whether the dialogue-based approaches effectively enhance the initial descriptions.

Table 4.3: Results on UCM dataset. Bold indicates the best result.

		Blip-2	**MLAT**	**Dialogue-based**		
				OED	**CFD**	**CFD-C**
Clipscore		65.2	69.0	66.1	**72.7**	67.8
ViT-L/14	R@1	21.9	4.8	**25.2**	22.9	9.0
	R@5	59.5	20.0	**61.0**	59.5	36.2
	R@10	83.8	37.1	**84.8**	81.4	63.3
T2IC	cos_sim	0.59	0.59	**0.60**	0.59	0.58
	FID	260.68	241.40	**238.71**	248.33	250.28

Descriptions obtained through Closed-Form Dialogue (CFD) achieve the highest CLIPScore, while those obtained with Open-Ended Dialogue (OED) achieve the highest text-to-image recall scores. This can be explained by the different approaches taken by OED and CFD. In CFD, questions are not affected by a wrong initial description because the questions are predetermined. For this reason, CFD can be more reliable in cases where the initial description is inaccurate. Conversely, the recall metric favors targeted descriptions, as greater specificity enhances the ability to distinguish an image from others. The OED approach achieves higher recall scores because it does not rely on a predefined template of questions, allowing for greater flexibility to tailor the dialogue to the actual image contents by formulating questions based on the first "plain" caption. CFD-C achieves the lowest recall scores; however, it ranks between CFD and OED based on the CLIPScore. Using the context inside the dialogue history to generate an answer does not improve the performance. The training using weak annotations is probably ineffective since CFD-C replicates errors produced in CFD. We expect that by training it on correct annotations,

Table 4.4: Results on RSICD dataset. Bold indicates the best result.

		Blip-2	MLAT	Dialogue-based		
				OED	CFD	CFD-C
Clipscore		65.8	71.1	66.1	**70.1**	67.8
ViT-L/14	R@1	9.2	2.56	**11.0**	6.7	3.9
	R@5	29.4	12.53	**32.7**	24.4	17.2
	R@10	44.3	21.68	**49.3**	38.6	28.5
T2IC	cos_sim	0.62	0.62	**0.63**	**0.63**	0.62
	FID	140.02	151.23	**122.03**	122.89	128.65

CFD-C can achieve higher performance by leveraging the previous context to predict the next answer. Some qualitative results for both datasets are shown in Figure 4.8 and Figure 4.9, where it can be noticed how the dialogues in the OED method are more targeted to explore further details of the initial "plain" caption. However, some questions are too specific and thus receive undefined answers. The CFD and CFD-C methods use more general questions and receive more general answers. In the third example, the network hallucinates the presence of an aircraft, highlighting the necessity of a correct and coherent predefined set of questions for closed-form dialoguing. The T2IC score comparing the original images with those reconstructed from the generated descriptions indicates that OED outperforms the others in terms of FID score. This suggests that summaries generated through dialogue are indirectly superior, enabling the text-to-image generation model to produce images more similar to the originals. However, based on the T2IC score for the RSICD dataset, it is difficult to determine the best method, as OED and CFD exhibit similar performance. We argue that this conclusion is more reliable, given that the RSICD dataset is larger than the UCM dataset, thus providing more robust statistics. While promising, this measure requires further refinement to enhance its reliability in this context. Specifically, more specialized image generation algorithms are needed to better address the remote sensing scenario. Figure 4.7 presents examples of both real and synthetic images, highlighting how dialogue-generated summaries capture more information regarding color, object count, and spatial distribution.

4.4 Discussion and conclusion

In this chapter, we introduced and analyzed three approaches for generating detailed textual descriptions of remote-sensing images using the concept of visual dialogue. Our proposed methodology leverages the instruction-following capabilities of recent large language models (LLMs) and vision-language models (VLMs, commonly referred to as "visual assistants") to establish a machine-to-machine visual dialogue (M2MVD). In this framework, a first machine sequentially questions a second machine about the image's contents. The second machine answers the questions, creating an artificial dialogue to explore the image. Questions can be generated using an LLM that autonomously decides what to ask based on an initial "plain" caption or predefined in a set of questions and identically applied to every input image. After the dialogue concludes, an LLM summarizes it into a paragraph, resulting in the final image description. Without ground truth for comparison, we propose three reference-free metrics to evaluate the coherence of the generated descriptions. Our results indicate that the Open-Ended Dialogue (OED) approach yields the most promising outcomes among the three methods. Nevertheless, we also discuss and analyze potential use cases for the Closed-Form Dialogue (CFD) approach. Despite the encouraging results, several limitations must be addressed to develop a more robust and customized pipeline. In the subsequent

section, we outline potential avenues for future research.

4.4.1 Image integration in the question generation process

In this work, the conditioning of the dialogue on the input image is limited to the answering process. Specifically, the large language model (LLM) responsible for generating the questions (ChatGPT) relies solely on the textual exchange to formulate subsequent questions. This approach can significantly impact performance, particularly when the initial description is inaccurate. Although Closed-Form approaches somewhat mitigate this issue, we contend that a more seamless integration of visual information into the dialogue process is crucial. Such integration could enhance the overall system by ensuring that the questions are more precisely targeted to the image content, thereby improving the quality and relevance of the generated descriptions.

4.4.2 Adaptation to the remote sensing context

Both machines employed in our dialogue-based approaches are not specifically targeted to remote sensing queries, which can lead to suboptimal performance. The answerer machine, in particular, is the most critical component since errors in prior answers can influence the formulation of subsequent questions. On the contrary, the questioner machine relies solely on textual information, thus requiring less adaptation since LLMs have already been trained on massive amounts of textual data. To mitigate this issue, remote sensing visual question-answering (RS-VQA) datasets can be utilized to refine the answer-generation algorithm and target it to the remote sensing domain. Also, using a custom image captioning model to obtain the initial "plain" caption can be beneficial, but at the cost of additional memory requirements to store the additional model. Our proposed CFD-C should not be viewed as a fine-tuned version of the system, as it leverages CFD results as weak training labels rather than human-annotated ones.

4.4.3 Removal of uncertain answers

Using the uncertainty prompt in Blip-2, numerous questions, particularly in Open-Ended Dialogue, result in answers such as "I don't know" or "not sure." The authors in [131] addressed this issue by implementing a specific ChatGPT prompt to minimize the inclusion of redundant and uninformative responses in the generated summary. However, this approach proves less effective in our context. We argue that the authors in [131] may not have fully recognized the significance of this challenge, as their dataset likely contained fewer such uncertain responses, given that their scenario involved natural images where Blip-2 is more effective. A more straightforward solution to mitigate this issue would be to exclude all question-answer pairs with uncertain responses before proceeding to the summarization stage.

4.4.4 Improved template questions

We observed that the formulation of certain questions for the CFD method presented limitations. An example is the question, "Which among vehicles, aircraft, or ships does the image contain, if any?". We observed that in cases where none of these objects are present, the question tends to receive erroneous interpretations from Blip-2. To address this issue, it is beneficial to develop more specific question templates that encourage the generation of more precise and relevant answers, particularly in contexts where the user already has a clear idea of the information they wish to extract.

4.4.5 Evaluation metrics

The evaluation strategies employed in this study rely on various reference-free metrics, which offer indirect assessments of the quality of generated dialogues. While these metrics provide valuable insights, they do

not directly measure the consistency of the generated descriptions to some defined ground truth. Ground-truth summaries produced by human annotators would be required to evaluate this aspect thoroughly. Simultaneously, further research into alternative reference-free metrics is essential, as these can provide a promising and more time-efficient evaluation method.

4.4.6 Customizing the Dialogue through prompting

Another aspect worth exploring is targeting the dialogue to meet the user's needs. As explored in this work, large language models adopt prompts to focus their attention on specific outputs the user needs. The Closed-form dialogue approach allows the user to target the dialogue by defining a set of questions, but this requires time and expertise. If the user is interested in certain details, it would be beneficial to have a prompt that can automatically target the questioner in generating questions to explore such details. Finding the best way to introduce such conditioning in the question-generation process can be a major advancement. Practically, this approach can be seen as an automatic generalization of CFD, in which instead of fixing a priori the questions, the user directs the questioner in producing a set of questions targeted to the need of the moment, alleviating the burden of the question definition.

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4.7: Original (**a**) and synthetic images generated starting from descriptions generated with different methods. (**b**) MLAT, (**c**) Blip-2, (**d**) OED, (**e**) CFD, and (**f**) CFD-C.

Figure 4.8: Examples of dialogues and summaries on RSICD images (**top row**) and UCM-Captions images (**bottom row**).

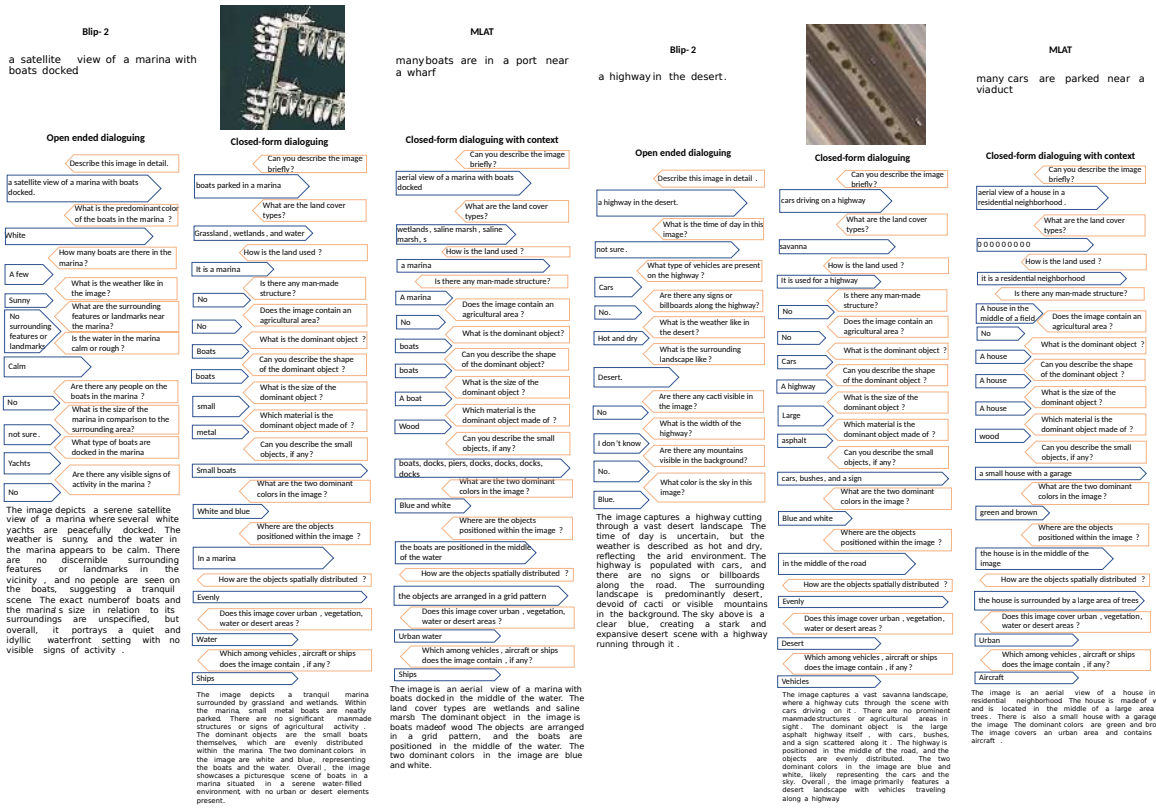


Figure 4.9: Examples of dialogues and summaries on RSICD images (**top row**) and UCM-Captions images (**bottom row**).

Chapter 5

Question Generation

[P4] Laila Bashmal, Yakoub Bazi, Farid Melgani, Riccardo Ricci, Mohamad M Al Rahhal, and Mansour Zuair. "Visual question generation from remote sensing images". In: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 16 (2023), pp. 3279–3293 (my contribution: conceptualization and discussion about results).

5.1 Introduction

The ability to generate coherent questions is as critical as the capacity to provide coherent answers. Over the years, researchers have extensively studied the problem of Visual Question Answering (VQA), developing systems capable of answering natural language questions about image content. VQA facilitates seamless interaction between users and systems, enabling users to pose questions in natural language and receive responses in the same format. However, existing VQA frameworks typically assume that human users generate the questions. This raises an important question: is it possible to empower machines with the capability to autonomously generate such questions?

The human ability to formulate informative and coherent questions plays a crucial role in acquiring new knowledge and shaping our understanding of the world. Enabling machines to perform this task holds significant potential, particularly in interactive systems. By equipping machines with the capability to generate questions, they can mimic human behavior and actively seek new information.

As discussed in the previous chapter, this capability can also be leveraged to enhance the question-generation process in our Machine-to-Machine Visual Dialogue (M2MVD) system. Our initial approach addressed question generation using a purely text-based methodology. Specifically, a large language model (LLM) utilized the initial image caption and subsequent dialogue as contextual input to generate the next question. This approach, which we term an indirect method of exploring visual content, relies solely on textual descriptions of the image rather than direct visual input. Consequently, the model "imagines" the image based on the provided text. However, as highlighted in the previous chapter, this formulation can lead to challenges, such as the dialogue diverging into irrelevant or non-meaningful exploration if the initial description lacks detail or accuracy.

By contrast, visual Question Generation (VQG) seeks to develop models capable of directly generating relevant questions by analyzing the visual content of an image. This direct interaction with the image mitigates the issues inherent in text-only approaches to visual dialogue. Despite its potential, research on VQG methods in remote sensing has been limited, with most efforts relying on template-based approaches, such as those employed in creating datasets like [109].

This chapter presents our work on open-ended visual question generation (VQG) for remote sensing scenes. The term open-ended refers to the capability to generate questions without relying on predefined

templates, instead predicting the words that compose the question in an auto-regressive manner. A primary challenge in remote sensing VQG is the lack of high-quality datasets tailored for this task, necessitating existing visual question-answering (VQA) datasets. However, VQA datasets are inherently answer-centric, exhibiting limitations such as reduced variability and naturalness in the questions, which pose significant drawbacks for question generation. These datasets often rely on template-generated questions, resulting in substantial redundancy. For instance, in the RSIVQA-DOTA dataset, only 32 unique questions are found among a total of 16430.

Moreover, remote sensing scenes are typically complex and rich in diverse concepts, requiring an effective VQG model to balance comprehensive concept coverage and question coherence. We introduce a novel VQA dataset, TextRS-VQA, specifically designed for visual question generation (VQG) and visual question answering (VQA) to address these limitations. TextRS-VQA is manually annotated to reduce redundancy and to include more natural, expressive questions. This chapter provides an overview of the dataset construction process, our methodology for VQG, and the results of our approach, emphasizing the challenges and opportunities in advancing this research area.

5.2 Metholodogy

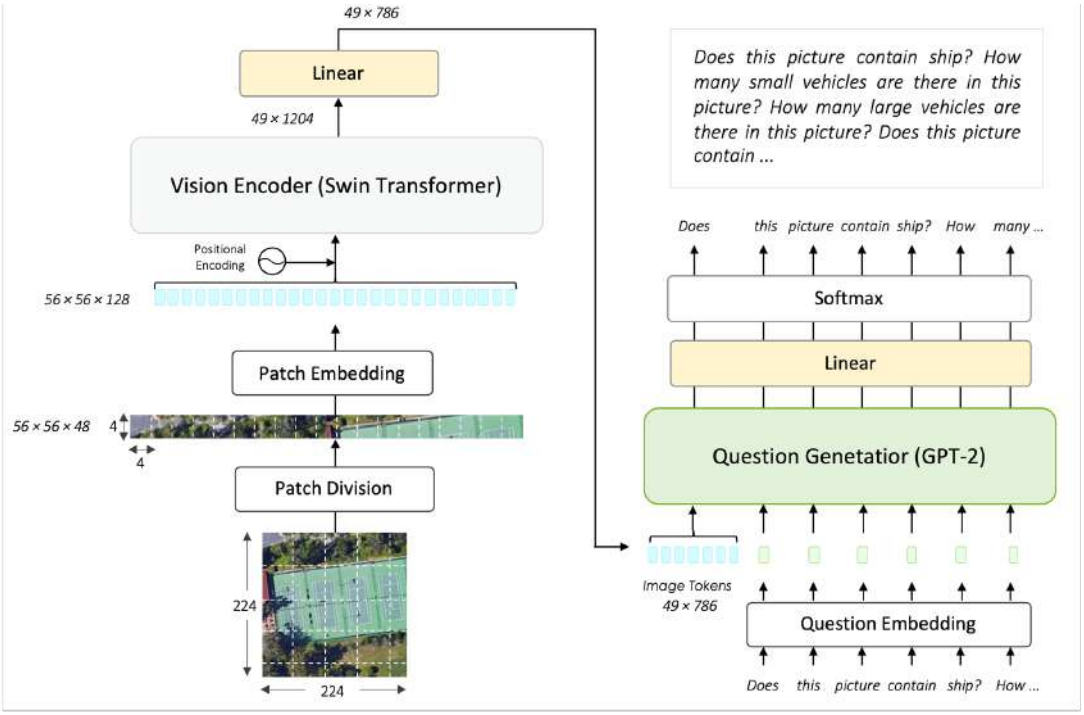


Figure 5.1: Overview of our VQG model. A vision encoder (Swin Transformer) extracts visual embeddings $\mathrm{Z}^x \in \mathbb{R}^{N \times d_x}$, which are projected using a linear layer to match the language model embedding dimension $d_t = 786$ and concatenated with the question embeddings to condition the text decoder (GPT-2) to directly generate the whole set of questions.

We envisioned a paragraph-based model that directly generates a set of questions. Specifically, a set of visual embeddings is concatenated to the stream of textual embeddings resulting from the set of

questions and fed to a language model decoder to guide the generation of the questions. We decided to use this formulation for two reasons linked to our previous work on visual dialogue. The first is that by directly generating an entire set of questions, we remove the need for the stopping criteria for visual dialogue. The second is that generating a set of questions directly is faster due to the generation step occurring only once at the beginning instead of at every step. However, directly generating a set of questions removes the dependency of successive questions on the answers to previous questions, which can result in repetitions or contradictions of previous "facts." Our paragraph-based question generation model works as follows. Let $\{\mathbf{X}_i, \mathbf{Q}_i\}_{i=1}^{N}$ be the i-th entry of a dataset composed of N samples. Each image $\mathbf{X}_i$ is associated with one or more questions $\mathbf{Q}_i = \{\mathbf{Q}_{i,j}\}_{j=1}^{J}$ where J is the number of questions associated with the image $\mathbf{X}_i$. We concatenate the set of J questions to derive the question paragraph $\mathbf{T}_i$, which is the desired response of the model subject to image $\mathbf{X}_i$.

$$\mathbf{T}_i = \mathbf{Q}_{i,1};\ \mathbf{Q}_{i,2}; \ldots ; \mathbf{Q}_{i,J} \tag{5.1}$$

Our paragraph-based VQG aims to directly generate the entire set of questions $\mathbf{T}_i$ given an RS image $\mathbf{X}_i$. The overall architecture of the VQG model is illustrated in Figure 5.1.

5.2.1 Vision Encoder

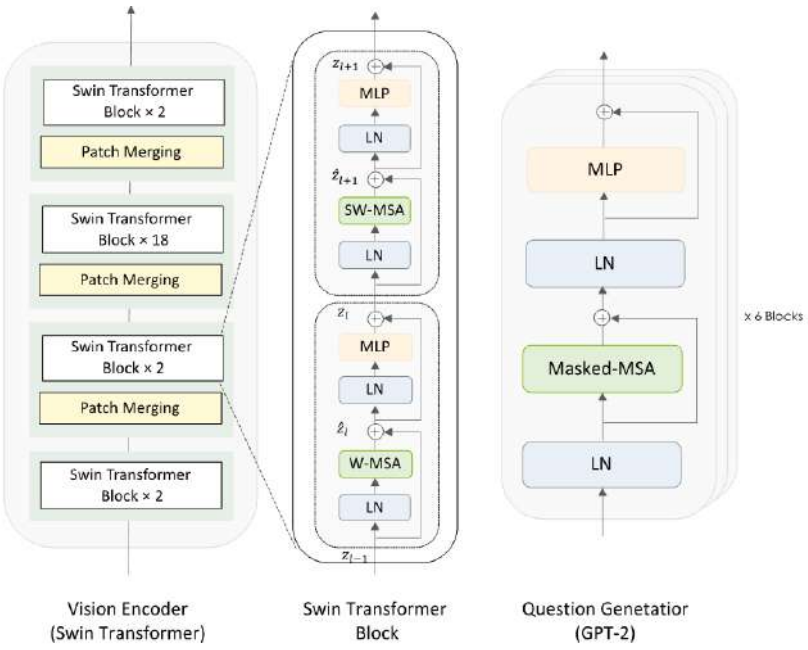


Figure 5.2: The internal architecture of the Vision Encoder, Swin Transformer Block, and GPT-2 block.

The initial step in our Visual Question Generation (VQG) model involves encoding the remote sensing (RS) image into a set of representative visual embeddings. For this task, we employ a variant of the Vision Transformer architecture, specifically the Swin Transformer [136]. This choice is motivated by the Swin Transformer's hierarchical architecture, which enables it to effectively extract global multi-scale representations from the RS image, capturing visual elements across varying scales. This is especially useful in RS images, characterized by objects spanning different orders of magnitude and granularity. The Swin Transformer condenses an image $\mathbf{X}$ into a sequence of N visual embeddings $\mathrm{Z}^x \in \mathbb{R}^{N,d_x}$. Specifically,

the process starts with an image $\mathbf{X} \in R^{224\times224\times3}$. The image is first divided into tiny non-overlapping patches of size of $4\times4\times3$, which are flattened to form a sequence of dimension $56\times56\times48$. The sequence is then projected by a patch embedding layer into the vision encoder dimension to form a feature of size $56\times56\times C$, where $C = 128$ in the implementation employed in our work. Positional information is added to the sequence before feeding it to a series of Swin Transformer layers.

Figure 5.2-a shows the vision encoder consisting of multiple Swin Transformer blocks interleaved with patch merging layers. The swin transformer layers alternate two special versions of self-attention called Window-based Self-Attention (W-MSA) and Shifter-Window-based Self-Attention (SW-MSA). W-MSA computes self-attention locally within each window, while SW-MSA shifts the features before partitioning and computing the attention. The local computation within the window employed by the first configuration reduces the computational requirements of the self-attention mechanism. In contrast, the shifted window-based self-attention allows the modeling of cross-window dependencies. The patch merging layer reduces the tokens between blocks by merging adjacent patches. Both configurations of the MSA use the self-attention mechanism, as described in the introduction. To summarize, each Swin Transform block alternates the computation as:

$$\begin{aligned} \hat{z}_l &= \text{WMSA}(\text{LN}(z_{l-1})) + z_{l-1} \\ z_l &= \text{MLP}(\text{LN}(\hat{z}_l)) + \hat{z}_l \\ \hat{z}_{l+1} &= \text{SWMSA}(\text{LN}(z_l)) + z_l \\ z_{l+1} &= \text{MLP}(\text{LN}(\hat{z}_{l+1})) + \hat{z}_{l+1} \end{aligned} \tag{5.2}$$

We use the output visual embedding sequence of the last block before the average pooling layer as the image feature representation $\mathrm{Z}^x \in \mathbb{R}^{N\times d_x}$, where $N = 49$ and $d_x = 1024$. To adapt the visual embedding dimension to the text embedding dimension of GPT-2 $d_t = 768$, we use a fully connected layer, projecting the visual feature to the desired dimension $\mathrm{Z}^{x_p} \in \mathbb{R}^{N\times d_t}$. The projected visual embedding sequence Z^{x_p} is fed into the language model as a prefix for generating the questions.

5.2.2 Language model decoder

We employ the pre-trained GPT-2 [62] as the language model to generate a set of questions for a given input image. The main motivation to use the pre-trained GPT-2 model is its effectiveness in data-limited scenarios due to the knowledge acquired through pre-training on a large-scale corpus of 8 million web pages and more than 1.5 billion tokens. We fine-tune GPT-2 on our dataset, transferring its learned linguistic knowledge to the RS questions generation task. We adopted a distilled version of the GPT-2, DistilGPT2, which has six identical transformer blocks. The DistilGPT2 can take up to 1024 token length, has 12 attention heads, and 82M parameters (compared to 124M parameters for the original GPT-2). The internal architecture of a single GPT-2 block is shown in Figure 5.2, which uses masked-multi-head self-attention (Masked-MSA), a variant of the attention explained in the introduction, that models auto-regressive prediction by masking future tokens from influencing the present token.

5.2.3 Model Training

During training, the goal is to fine-tune the parameters of the questions generator (GPT-2) to maximize the probability of generating the question paragraph $\mathbf{T}_i$ for an input image $\mathbf{X}_i$. The objective of the question generator is the auto-regressive language modeling objective function described in equation 2.4, conditioned by the sequence of visual embeddings Z^{x_p}. We use greedy sampling to choose the highest probable token as the next token in the predicted sequence.

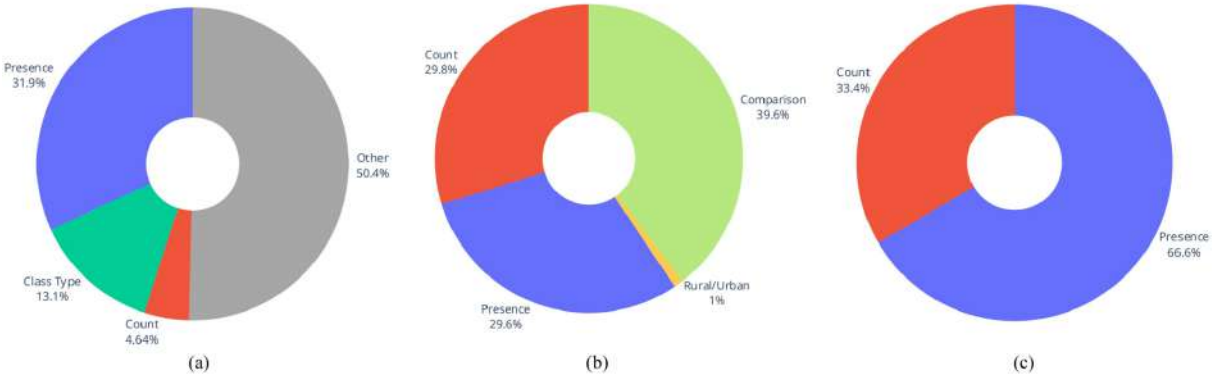


Figure 5.3: Distribution of question type for each dataset: (a) TextRS-VQA, (b) RSVQA-LR, (c) RSIVQA-DOTA

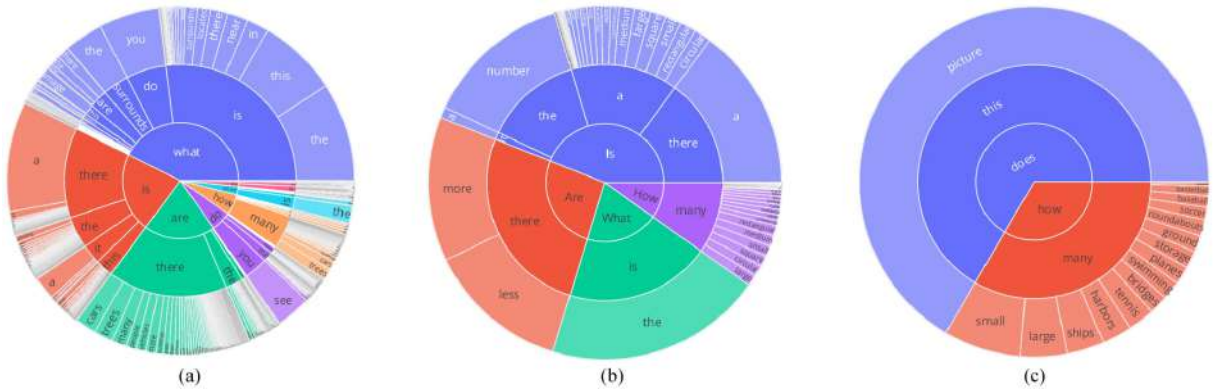


Figure 5.4: Sunburst visualization for each dataset and question type: (a) TextRS-VQA, (b) RSVQA-LR, (c) RSIVQA-DOTA

5.3 TextRS-VQA Dataset

VQG has made significant progress in computer vision. A key reason for this progress is the availability of large datasets with high-quality questions of various types [137][138]. However, in the RS domain, development is still limited, and the existing datasets, such as RSVQA [125] and RSIVQA [110], which were proposed to solve the VQA problem, are more focused on the answers rather than the questions. Questions in those datasets are generic, repetitive, and lack diversity, as they are automatically generated from predefined templates. Thus, it is essential to establish a suitable benchmark to advance the research on RS visual question generation. To fill this gap, we construct the TextRS-VQA dataset, the first fully manually annotated dataset for VQA and VQG tasks in the RS domain. This dataset addresses the main problem of prior datasets, which is the high redundancy of the questions. You can access our dataset at this link. Figure 5.5 shows some sample images and their associated questions from our datasets and previously proposed VQA datasets in RS. It can be seen how the questions in our dataset are more natural, less redundant, and highly relevant to the image contents. Further, Table 5.1 summarizes some statistics from our TextRS-VQA dataset and the other two VQA datasets in RS. The statistics show that although our dataset has the lowest questions per image rate, the *diversity* of these questions –

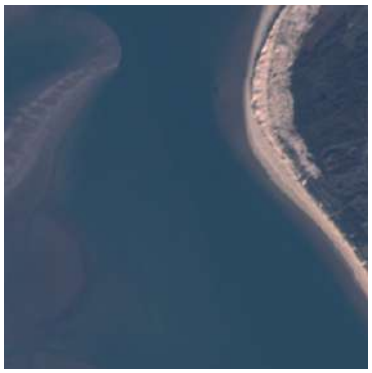
RSVQA-LR	**RSIVQA**	**TextiRS**
		
- Is it a rural or an urban area? - Is the number of commercial buildings equal to the number of commercial buildings? - Is there a farmland?	- How many small vehicles are there in this picture? - Does this picture contain a small vehicle? - Does this picture contain a plane?	- Is there aircraft next to the loading area? - What is the green area next to the terminal? - How many aircraft are in this terminal?
		
- Is a commercial building present? - Is there a building? - What is the number of roads?	- How many baseball diamonds are there in this picture? - Does this picture contain a baseball diamond? - Does this picture contain a harbor?	- How many basketball courts are in the image? - Are there buildings in the image? - Is there a green passage near the basketball courts?

Figure 5.5: Comparison of question types in RSVQA-LR, RSIVQA, and TextiRS datasets.

the number of unique questions to the total number of questions – is higher. In addition, our dataset has a lower question average length but richer vocabulary compared to the other datasets. Figure 5.3 visualizes the distribution of the question types for the three datasets. As can be seen from the figure, RSVQA-LR has four question types, and RSIVQA-DOTA has only two types of questions. In contrast, in our TextRS-VQA dataset, around half of the questions are from the "presence," "class type," and "count" types, while the other half contains a mix of other types. To further illustrate the diversity of our dataset, Figure 5.4 provides a sunburst visualization of the question's first three words for each dataset. As can be seen, our TextRS-VQA revealed greater diversity than other VQA datasets.

5.3.1 Question annotation

The images of our TextRS-VQA dataset come from the TextRS dataset, which was introduced for RS image retrieval and captioning [139]. It was built by collecting images from four well-known scene classification datasets with different image sizes and spatial resolutions. Namely, AID [140], PatternNet [141], UC-Merced [142], and RESISC45 [143]. From each dataset, 16 images are randomly extracted for every class, totaling 2144 images. Each image is manually annotated with two to five questions to ensure diversity, focusing on questions tailored to RS use cases. Our dataset contains 6245 questions of four types: object counting, presence/absence, scene classification, and "other," which includes a wide range of questions. We randomly selected 80% of the images and their associated questions for training while the remaining 20% are left for testing.

Table 5.1: Statistics comparison between remote sensing VQA Datasets. Diversity is measured as the ratio of images with at least a unique question over the total number of images. Unique means that the question is not associated to any other image

Dataset	**TextRS-VQA**	**RSVQA-LR**	**RSIVQA-DOTA**
# Images	2143	772	1868
# Questions	6245	77232	16430
# Unique questions	3608	8206	32
Diversity	57.77%	10.62%	0.002%
# Questions per image	2–5	100–101	3–24
Avg. question length (in words)	6.62	8	6.67
# Unique words in questions	609	131	47

5.4 Results

In this section, we first give the experimental details of our VQG model and explain the adopted evaluation metrics before presenting the results.

5.4.1 Experimental setting

We used the base version of the Swin Transformer model as the vision backbone. The model has been pre-trained on a large dataset of natural images (ImageNet-22k) and then fine-tuned on ImageNet-1k, with images at 224×224 resolution. The model uses a patch size of 4×4 pixels and a window size of 7×7 patches. The model has an embedding dimension $d_x = 128$. The base model consists of 4 layers; the first, the second, and the last layers each have two Swin Transformer blocks, while the third layer has 18 blocks. Patch merging is applied after the first three blocks. For question generation, we use DistilGPT2, a

Table 5.2: Results when fine-tuning GPT-2. B1-B4 (Bleu-1 to Bleu-4)

Dataset	Question	B1	B2	B3	B4	ROUGE	METEOR	CIDEr
TextRS-VQA	1^{st}	56.40	42.77	31.81	24.32	49.27	22.43	85.64
	10^{th}	49.63	32.77	20.04	12.71	41.28	17.13	48.85
RSVQA-LR	1^{st}	100.0	100.0	100.0	100.0	100.0	100.0	00.00
	20^{th}	99.21	93.79	85.62	79.09	87.78	47.16	29.32
RSIVQA-DOTA	1^{st}	97.30	95.79	94.11	92.85	90.91	65.73	126.30
	20^{th}	77.34	74.31	70.67	65.20	73.46	46.89	28.74

Table 5.3: Results when freezing GPT-2. B1-B4 (Bleu-1 to Bleu-4)

Dataset	Question	B1	B2	B3	B4	ROUGE	METEOR	CIDEr
TextRS-VQA	1^{st}	59.93	45.37	32.96	24.40	51.59	22.83	87.45
	10^{th}	42.07	30.74	19.03	11.45	39.35	14.85	46.50
RSVQA-LR	1^{st}	100.0	100.0	100.0	100.0	100.0	100.0	0.00
	20^{th}	100.0	99.08	87.98	78.96	92.49	47.90	31.97
RSIVQA-DOTA	1^{st}	94.76	91.66	87.98	85.26	87.90	57.10	90.81
	20^{th}	77.53	74.76	71.18	65.72	74.05	47.87	33.64

smaller model distilled from GPT-2. This model can process sequences up to 1024 tokens and has 6 layers, 768-hidden feature representation, 12 attention heads, $82M$ parameters (compared to $124M$ parameters for GPT-2). We use the AdamW optimizer with a batch size of $bs = 16$ and fine-tune the model for 128 epochs. The initial learning rate is $2e-5$, which decays at 0.05. The number of questions to generate during inference is fixed based on the dataset. For RSVQA-LR and the RSIVQA-DOTA, a maximum of 20 is generated, while for TextRS-VQA, this limit is set to 10. Different evaluation metrics are used to measure the quality of the questions generated by our VQG model. As standard metrics, we employ BLEU, METEOR, CIDEr, and ROUGE, reference-based metrics that measure how well the generated questions match the ground-truth questions in the dataset. We use two other metrics, strength, and inventiveness, to evaluate the diversity of the generated questions. Strength is the percentage of unique questions generated per image, while inventiveness is the ratio of the unique generated questions never seen in the training set.

5.4.2 Numerical results

Since our model generates a paragraph of questions, we report the metrics for both the first and last question of the paragraph. Generally, standard metrics assign lower scores for results on TextRS-VQA compared to the other two datasets, especially regarding BLEU, ROUGE, and METEOR. This is due to the higher question diversity in TextRS-VQA, which results in a more challenging scenario for generative algorithms. However, this can also be due to issues not related to the generated text but to the evaluation criteria, which rely heavily on exact word matching. This is explored in [131], in which it is shown that standard metrics report drops in performance despite the semantic coherence of the output. Performance on TextRS-VQA demonstrates a decline of approximately 5-11% across nearly all metrics from the first to the last question. On RSVQA-LR, the first question receives perfect scores in all metrics except the CIDEr. This is because the generated questions perfectly match the ground-truth questions, and

the low CIDEr score is due to its high weight assigned to rare keywords, as well as the high degree of repetitiveness in this dataset. Generally, scores tend to decrease from the first to the last question. On RSIVQA-DOTA, the scores of almost all metrics are stable for the first six questions in the paragraph. However, the scores started to drop gradually, starting with question seven. We further analyzed the effect of freezing or fine-tuning the weights of the language model decoder. Table 5.3 shows the results of fixing the weights of the GPT-2 model. We can see that for the TextRS-VQA dataset, the scores for generating the first question are better when the language model weights are not updated. A hypothesis for this behavior is that freezing the language model decoder preserves some entropy on the output classification, leaving space for the model to generate more diverse questions and, thus, better match the distribution on our TextRS-VQA dataset. This is also confirmed by a generally higher CIDEr score in results from the frozen model. However, other metrics assign slightly higher scores to the last question generated by the fine-tuned language model. On the RSVQA-LR dataset, the results of fixing and fine-tuning the language model are identical to the first questions in the paragraph. However, the model with fixed weights scores better in almost all metrics as the model generates more questions. For RSIVQA-DOTA, the first generated question by the fine-tuned model is more accurate than the first generated question by the fixed model. However, scores of fixed and fine-tuned models are comparable when generating later questions.

Results of diversity metrics

To provide an in-depth analysis of the generated questions, Table 5.4 reports the generative strength and inventiveness of our model when the language model is fine-tuned or is kept fixed. In all datasets, the generative strength of the fine-tuned language model is better compared to the fixed one. In contrast, the question generator with fixed weights has high inventiveness scores and more ability to generate novel questions that do not exist in the training set. This is a further confirmation of the previous hypothesis that a fixed language model preserves some entropy in the output distribution. The results also show that the inventiveness of the model on the TextRS-VQA dataset is higher than other datasets despite our dataset having the least number of questions. This could be attributed to the richness and the high diversity of the questions in our dataset, which result in a more diverse and expressive model. In addition, the table shows that the inventiveness scores of the fine-tuned model on the RSVQA-LR and the RSIVQA-DOTA datasets are close to zero, which is an indication of the very low diversity of questions in the dataset.

Table 5.4: Results of Diversity Metrics

Dataset	Training Strategy	Strength	Inventiveness
TextRS-VQA	Fine-tuned	83.71	63.06
	Fixed	58.79	82.54
RSVQA-LR	Fine-tuned	95.95	0.007
	Fixed	45.65	9.68
RSVQA-DOTA	Fine-tuned	81.31	0
	Fixed	62.92	15.38

5.4.3 Qualitative Results

Figure 5.6 shows examples of generated questions from test images of each dataset, along with the ground-truth questions. From the TextRS-VQA example, we can notice that the questions generated by our model

	TextRS-VQA	**RSVQA-LR**	**RSVQA-DOTA**
			
Predicted	- Are there a building in front of the parking lot? - How many cars are in the parking lot? - Are there a building next to the parking lot? - Is this parking lot full of cars? - Is there a red car parked in the parking lot? ...	- Are there less commercial buildings than grass areas? - Are there less residential buildings than grass areas? - Are there less water areas than residential buildings? - Is the number of roads equal to the number? ...	- How many small vehicles are there in this picture? - How many roundabouts are there in this picture? - Does this picture contain small vehicles? - Does this picture contain a plane? ...
Ground-truth	- Are there lines on the ground of the parking lot? - Is the parking space all occupied? - Is there discoloration of the cars?	- Is it a rural or a urban area? - Is there a grass area? - What is the number of roads? - Is there a road? ...	- How many small vehicles are there in this picture? - How many roundabouts are there in this picture? - How many swimming pools are there in this picture? - Does this picture contain small vehicle? ...

Figure 5.6: Examples of questions generated by our VQG model for test images of each dataset.

contain almost all the semantic information present in the image. Although the generated questions do not perfectly match the ground-truth questions, some questions have the same meaning as the ground-truth questions but are expressed differently, such as the reference question "Is the parking space all occupied?" and the generated question "Is the parking lot full of cars?". Further, our model can generate novel and valid questions correlated with the image content but not present in ground-truth questions such as "How many cars are in the parking lot?". We can also see some questions about objects and their attributes that are not present in the ground-truth questions, such as questions about "buildings" and "red cars." This proves that our model can precisely understand image content and translate it into questions. Most of the questions in the example seem biased toward the presence question, and there are some incorrect questions, such as "Are there cars on the road empty?". In general, however, the questions are sensible, diverse, and in line with the semantic content of the image. The example of the RSVQA-LR

dataset shows that the generated questions are very close to the ground-truth questions, even though there are some incomplete questions, such as "Is the number of roads equal to the number?". In addition, all questions follow the same pattern and question types of ground-truth questions, which aligns with the quantitative results that show high similarity with the reference questions and low inventiveness scores. A similar conclusion can be drawn by observing the generated questions for RSIVQA-DOTA. All the generated questions are from the presence and count types, which are the only question types defined for this dataset, which also explains the low inventiveness of our model on this particular dataset.

5.4.4 Comparison with state of the art methods on MSCOCO dataset

To verify the performance of the proposed method, we compare it against several VQG models in computer vision on MS-COCO VQA dataset [144], which consists of 82783 train images and 40504 test images. Each image in this dataset is associated with 5.4 questions on average. The results are presented in Table 5.5. Generally, our VQG model outperforms all existing state-of-the-art methods regarding the four BLEU metrics and METEOR. Our model also achieves the second-best ROUGE score. However, the score of our model in terms of the CIDEr is lower than the state-of-the-art methods.

Table 5.5: Experimental results of our method and other state-of-the-art VQG methods on MS-COCO-VQA dataset. Best results highlighted in bold, second best results underlined.

Method	B1	B2	B3	B4	ROUGE	METEOR	CIDEr
IA2Q [145]	32.43	15.49	9.24	6.23	-	11.21	36.22
V-IA2Q [146]	36.91	17.79	10.21	6.25	-	12.39	36.39
Krishna et al. (t-space) [147]	47.40	28.95	19.93	14.49	49.10	18.35	<u>85.99</u>
Krishna et al. (z-space) [147]	<u>50.09</u>	<u>32.32</u>	<u>24.61</u>	<u>16.27</u>	-	**20.58**	**94.33**
IC2Q (WS) [145]	30.42	13.55	6.23	4.44	-	9.42	27.42
V-IC2Q (WS) [146]	35.40	25.55	14.94	10.78	-	13.35	42.54
Krishna et al. (WS) [147]	31.20	16.20	11.11	6.24	40.27	15.77	35.89
I + II + CL (WS) [148]	38.94	20.30	12.37	8.10	41.27	13.47	37.42
I + II + CL + Bayes (WS) [148]	41.87	22.11	14.12	10.04	42.34	13.63	46.87
Ours	**55.34**	**38.44**	**26.06**	**17.62**	**47.45**	<u>19.32</u>	29.63

WS: Weakly Supervised.

5.5 Conclusions

This chapter presents a novel dataset and methodology for remote sensing visual question generation (RS-VQG). This work addresses a critical gap in remote sensing VQG datasets, which are predominantly answer-centric and lack the diversity and naturalness of human-generated questions. Our model generates a paragraph comprising multiple questions but does not incorporate the sequential conditioning of questions on the answers to preceding questions. Future extensions of this work could involve conditioning the generation of successive questions on prior answers, which would require the development of a specialized dataset featuring question-answer pairs organized in a coherent flow.

The proposed approach employs an encoder-decoder architecture, leveraging a Swin Transformer to encode the image into a sequence of multi-scale embeddings and a GPT-2 model to auto-regressively decode a set of questions based on these visual embeddings. We evaluate the performance of our method on two standard visual question-answering (VQA) datasets and our newly proposed TextRS-VQA dataset.

The results demonstrate the effectiveness of our model in generating a diverse and meaningful set of questions from remote-sensing images. Additionally, our manually annotated TextRS-VQA dataset exhibits significantly higher question diversity, offering an appealing resource for enhancing the naturalness and creativity of question generation in RS-VQG tasks.

Chapter 6

Visual Dialogue for Change Captioning

[P6] Riccardo Ricci, Yakoub Bazi, and Farid Melgani. "Change Captioning Meets Large Language and Vision Models". In: Under revision for ISPRS Journal of Photogrammetry and Remote Sensing, 2024.

6.1 Introduction

Change captioning aims at describing through natural language changes between two (or more) views of the same location acquired at different times. In contrast to conventional change detection outputs such as semantic or binary change maps, natural language provides a quicker and more intuitive interpretation of the results for human users. Furthermore, natural language conveys richer semantics, enabling analysis of the relationships between changed areas, their scale, and their attributes. Initial change captioning research [149] focused on pairs of frames acquired by surveillance systems. The authors used a combination of cluster alignment and auto-regressive decoding to generate short descriptions of changes. In [150], the authors leveraged the CLEVR engine to build a new dataset of synthetic images for change captioning under the influence of "distractors." Distractors, such as viewpoint changes, are elements that introduce fictitious changes while the semantic content remains unchanged. The authors proposed a dual attention network to focus on changed areas and extract features. Successively, they input these features into an attention-based recurrent neural network (RNN) to describe the changes. The RNN decoder dynamically attends to features from the before, after, or different images as it generates the caption. With a similar goal, [151] introduced a novel M-VAM encoder that explicitly distinguishes semantic changes from viewpoint changes in the feature space. A differential encoder distinguishes semantic features from viewpoint changes in [152]. Furthermore, the authors incorporate a cycle consistency module that aligns the combined features of the caption and the first image with those of the second image, enhancing overall performance. Authors of [153] leveraged attention to capture semantic relations between different views dynamically. They employ a visual attention block and a visual switch during decoding to dynamically modulate the context (attending to the before, after, or difference image features) and its influence on caption decoding. In [154], the authors propose a multitask learning setting to improve performance by tackling change captioning and query image retrieval simultaneously using the generated caption. A different avenue has been explored in [155], in which the authors proposed a novel benchmark dataset with multiple changes between pairs of images. Transformer neural networks are employed to localize and describe all the changes.

Although mainly explored using synthetic datasets of natural images, change captioning can empower

several remote sensing applications such as urban planning, disaster management, and environmental monitoring and surveillance [156]. Remote sensing change captioning has been introduced in [157], where the authors employ a pre-trained CNN to extract visual embeddings from images. They explore various methods to fuse the embeddings of the image before and after the changes and use the fused embedding as a conditioning to generate the change caption. In another study [158], a novel transformer-based network is proposed to enhance change captioning performance. The authors further propose a new large-scale dataset and test several other approaches, with their method achieving state-of-the-art results. Authors in [159] utilize the attention mechanism embedded in a multi-scale interactive change-aware encoder. A multi-layer adaptive fusion module enables the encoder to capture semantic changes between bi-temporal image features effectively. Captions of changes are subsequently generated by a transformer decoder, enhanced by cross-gated attention to integrate visual information. In [160], the authors explicitly guide the model, feeding difference features to multiple progressive difference perception layers and enhancing spatial awareness of changes. Authors adopt a similar approach in [161], employing a "difference" feature encoding block. However, they utilize a single-stream feature extractor pre-trained on remote sensing images to minimize the domain gap and improve downstream performance. In [162], the authors propose a novel approach by decoupling change captioning into two distinct sub-tasks: a binary classification to determine the presence of change and the subsequent task of describing the change. They also employ prompt learning to harness the power of a pre-trained LLM, thereby maintaining low training memory requirements.

All the aforementioned change captioning research has focused on *supervised learning*, which requires the collection of large quantities of data in the form of triplets of two images and a text describing the changes $(\mathbf{X_1}, \mathbf{X_2}, \mathbf{T})$. The collection of large quantities of data is severely time and resource-consuming. Furthermore, different users may be interested in different changes, making the customization of the output a desired requisite for change captioning. In the conventional scenario, this would require a different dataset to cover each different change, severely impacting the feasibility of this paradigm. Another obstacle can be the training of large models, which is resource-intensive and not affordable for most users. Thus, our work explores the possibility of leveraging the pre-trained capabilities of instruction-following Large Language Models (LLM) and Vision-Language Models (VLM) to generate meaningful descriptions of the changes between pairs of images. We remove the need for training by designing methodologies to stimulate *pre-trained* models and elicit their intrinsic capabilities. We test prompting as an approach to adaptively steer the behavior of an LLM to analyze different changes. Additionally, we introduce and validate a novel evaluation criterion, FMScore (Fact Matching Score), which serves as a reference-free metric for assessing the quality of change descriptions by utilizing fact matching as a proxy for description accuracy.

6.2 Methodology

Motivated by the strong zero-shot performance of LLMs and VLMs, we employed two pre-trained models in our experiment. Similarly to our previous experiment on visual dialogue for image description in Chapter 2, we adopted Vicuna [163] for pure text-based tasks. Specifically, we use Vicuna-13b-v1.5, a 13B parameters large language model based on LLaMA-2, fine-tuned on conversational data to acquire instruction-following capabilities. For vision tasks, we employ OTTER-Image-MPT7B [164], a 9B parameters large vision-language model based on OpenFlamingo-9B, and fine-tuned on the MIMIC-IT dataset [165]. Otter is characterized by its ability to accept interleaved image and text instructions, making it feasible to work with sequences of images, as described by the authors.

We developed three distinct approaches for extracting remote sensing change descriptions, each characterized by varying levels of complexity and potential for customization. In the first approach, the most straightforward, we directly prompt Otter to compare the two images and describe the changes.

In the second approach, we simplify the task by dividing it into two sub-tasks. Using Otter, we extract separate descriptions for each image. Then, leveraging Vicuna, we compare the textual descriptions to infer potential changes. The third approach is based on the concept of dialogue, introduced in Chapter 2. Instead of directly generating a change description, we use a series of question-answer pairs to extract specific information step-by-step. We explore two variations of dialogue: open and template. In open dialogue, Vicuna is instructed to generate questions that explore potential changes. Conversely, in template dialogue, a predefined set of questions is sequentially posed to Otter for each pair of images. In both versions, Otter provides answers conditioned on the question and the visual information (pair of images). More in detail, given a pair of multi-temporal satellite images, $\mathbf{X_1}$ and $\mathbf{X_2}$, Otter first extracts visual embeddings Z_x using CLIP [91], then uses a Perceiver Resampler module to compress the information in the sequence of visual tokens into a small fixed number of learnable queries, effectively acting as an information bottleneck. The "perceived" visual tokens are exploited as keys and values of several cross-attention layers inside the language model.

6.2.1 Direct change paragraph extraction

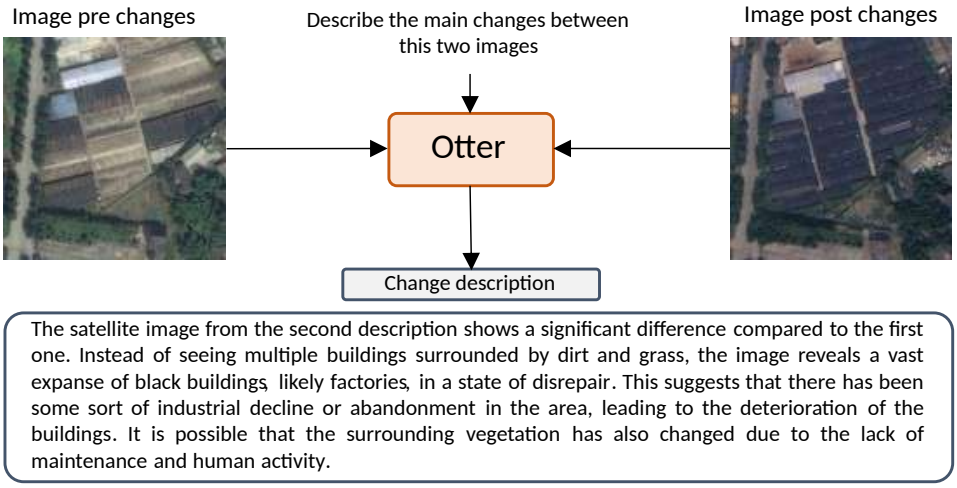


Figure 6.1: In direct extraction, Otter is directly fed with the pair of pre and post-change images ($\mathbf{X_1}$,$\mathbf{X_2}$) and prompted with a targeted instruction (see Tab. 6.6), to which Otter answers with the change description.

Figure 6.1 depicts the scheme of our first proposal. Otter, conditioned on the visual content and stimulated with a targeted instruction (see Table 6.6), directly creates a paragraph describing the changes. Despite the simplicity of the approach, a substantial drawback is the non-specificity of the generated paragraph, which is general and may not fit the user's needs.

6.2.2 Indirect change paragraph extraction

Directly extracting a paragraph describing the changes is challenging and unconventional, particularly since Otter is not specifically designed for remote sensing applications. Therefore in our second approach, we decompose the process into two more manageable tasks. The scheme is depicted in Figure 6.2. First, Otter is instructed to describe each image separately. Successively, Vicuna analyzes the two textual descriptions to extrapolate the changes, yielding the final change description. For the second step, we

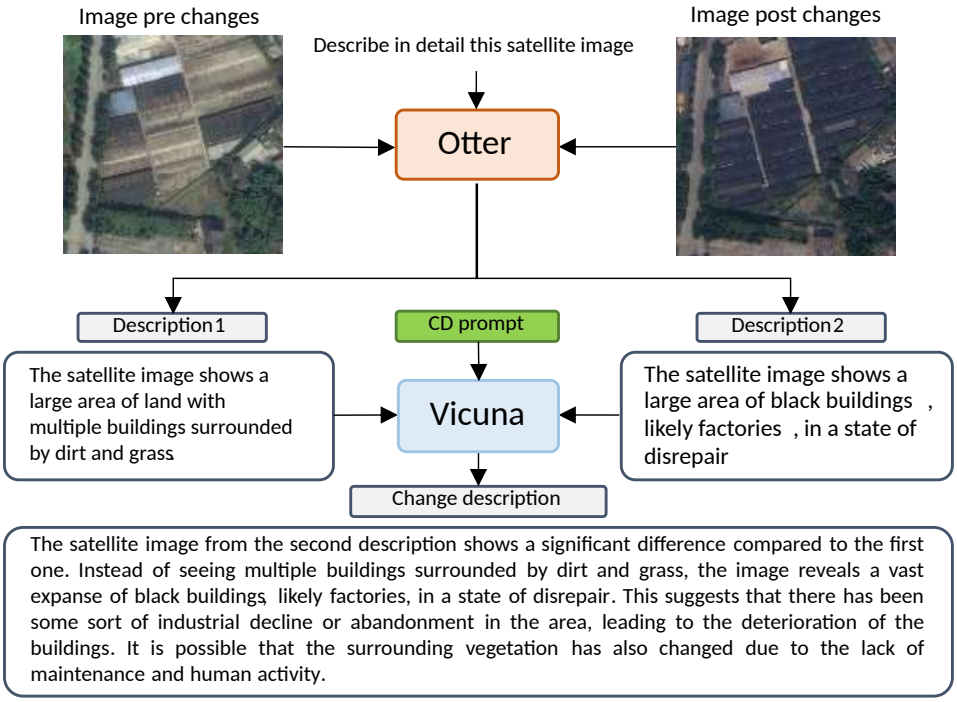


Figure 6.2: Indirect extraction has Otter describing each image separately. Then Vicuna analyzes the two descriptions to infer possible changes based solely on the text, generating the final change description.

decided to use Vicuna for its excellence in text comprehension tasks. See Table 6.6 to inspect the prompts we used for each step and model. The decomposition of the process can reduce the task difficulty but can raise two main problems. The first is error accumulation; a two-step approach will inevitably accumulate mistakes from two generative steps. The second is the target agnosticism of the first step, where the model can produce descriptions of different aspects of the image, impacting the extraction of the real changes between the images.

6.2.3 Dialogue-based change paragraph extraction

The third approach, depicted in Figure 6.3, addresses the limitations of the first two. Both first and second approaches are insensitive to user needs and cannot be efficiently targeted to specific aspects of interest. Toward change description customization, we leverage the concept of dialogue to extract small chunks of information sequentially. By carefully formulating targeted questions, the model is constrained to narrow requests, thereby reducing the likelihood of deviations from the content of interest to the user. Leveraging our work on visual dialogue, we designed two complementary paths, *template* and *open* dialogue. In template dialogue, as described in Chapter 2, the user defines a fixed set of questions for all the pairs of images (refer to the Appendix for the template set of questions used in this study). In open dialogue, a language model (in our case, Vicuna) automatically generates questions, ingesting the previous dialogue history. In this formulation, the user can optionally define its needs in broad terms during prompting to steer the generation toward a particular aspect of interest (see the experiment in Section 6.6.3). After the dialogue, the conversation is summarized using Vicuna, yielding the final change description.

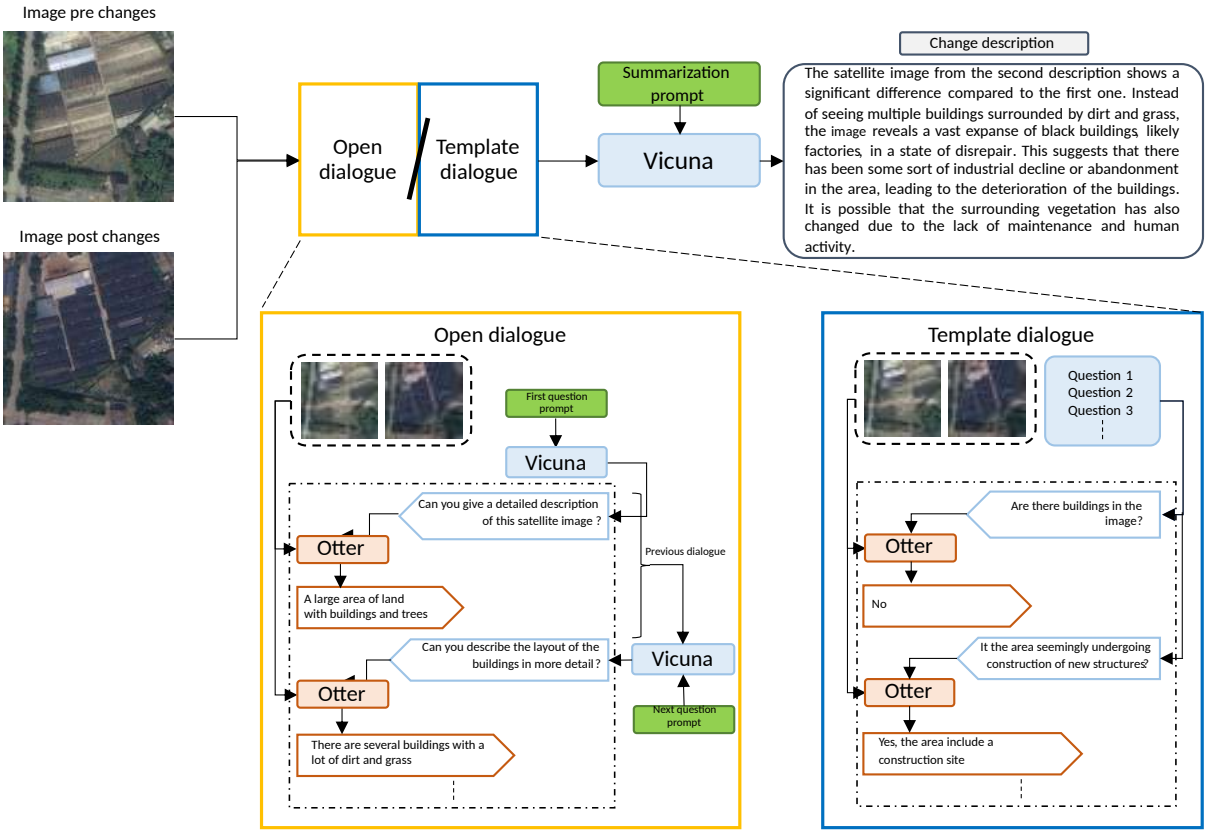


Figure 6.3: In dialogue-based change paragraph extraction, a simulated conversation is used to collect information. Then, Vicuna is instructed to summarize the dialogue and create the final change description.

6.3 Datasets

We test our approaches on two datasets, SECOND and LEVIR-CC.

SECOND [166] contains 2968 pairs of aerial images acquired over Hangzhou, Chengdu, and Shanghai. Each image is of size 512×512 and is annotated with a semantic change map at the pixel level. The images show changes in 6 land-cover classes, namely *non-vegetated ground surface*, *tree*, *low vegetation*, *water*, *buildings*, and *playgrounds*. We adopted an augmented version of this dataset [167], which contains several question-answer pairs for each couple of images created with an automatic pipeline, as depicted in Figure 6.5.

LEVIR-CC [158] is specifically designed for remote sensing change captioning tasks. It comprises $10,077$ pairs of bi-temporal images and $50,385$ sentences describing the observed differences. Each image pair is annotated with 5 captions to provide diverse perspectives on the changes detected. All the images are standardized to a size of 256×256 pixels and are registered at the pixel level to mitigate variations caused by viewpoint changes. Some examples are depicted in figure 6.4.

(a) Before (Pre-Event)

(b) After (Post-Event)

Change Captions
1. Some completed houses are lined by the road.
2. Three neatly arranged buildings with paths to the road are built.
3. Several houses are constructed on the bare land on one side of the road.
4. Three houses are built along the road.
5. Three houses show up along the road.

Figure 6.4: Multitemporal images from LEVIR-CC with ground truth change captions.

6.4 Evaluation

Evaluating change paragraphs is difficult for two reasons. The first problem is obtaining a good ground truth, given the costs in terms of time and effort. There exist datasets containing ground truth annotations, like [158], but in the form of single sentence descriptions, which are very different from our context of longer and more articulated *paragraphs*. The second reason is that standard natural language processing (NLP) metrics are designed to evaluate the quality of short sentences. Synonyms, paraphrases, and rearrangement of words are not taken into consideration by standard metrics, which mainly prioritize exact word matching over semantic content. Since we aim to extract a description that can take different syntactic forms, a metric isolating semantics from syntax could be very helpful. For this reason, we decided to complement standard metrics by proposing a novel LLM-based evaluation method, FMScore, to evaluate long paragraphs using an exact "fact" matching evaluation.

We adopt four standard NLP evaluation metrics, namely BLEU, METEOR, CIDEr, and ROUGE-L. To obtain references on SECOND, we created pseudo-ground-truth paragraphs using ChatGPT, prompting it to summarize the concatenations of all the questions and answers associated with each pair of images. It is worth noting that given the particularity of the questions and answers, the resulting paragraphs are rich in specific words, such as class names or acronyms of the classes. On LEVIR-CC, we used ground truth change captions as the reference for standard metrics.

6.4.1 LLM-based evaluation - FMScore

To design a general-purpose metric for evaluating change paragraphs, we envisioned a novel strategy that uses an LLM to indirectly assess the coherence of the generated change descriptions. The hypothesis is that a good change description should contain references to all the changes between the two images while minimizing the references to spurious changes. Reformulating, given two images, we should aim at

(a) Before (Pre-Event)

(b) After (Post-Event)

Questions	**Answers**
Have the regions of non-vegetated ground surface changed?	Yes
Have the regions of buildings changed in the first image?	Yes
What is the largest change?	Buildings
What is the percentage of changed areas?	60_to_70
Did the regions of water decrease?	No
...	...
Summary	
The two multitemporal images show changes in different land cover types. The areas of non-vegetated ground surface, trees, and low vegetation have changed in both images. The areas of water and playgrounds have remained mostly unchanged. The regions of buildings have increased in both images, with the largest change in the post-event image. The change proportion of non-vegetated ground surface in the first image is higher than in the post-change image. The change proportions of trees, low vegetation, and water in the second image are low.	

Figure 6.5: Multitemporal images from SECOND with questions, answers, and summary. (Bottom) LEVIR-CC dataset samples with ground truth captions.

creating a description that makes it possible to "deduce" all and only the changes that happened. The potential of this evaluation method is that it replaces exact word matching with exact "fact" matching. We think a technique based on this assumption can provide better insights compared to standard metrics, especially for long and articulated paragraphs. In the subsequent section, we elaborate on our concept and provide a comprehensive overview of our metric.

6.5 FMScore (Fact Matching Score)

FMScore is a metric that indirectly measures the quality of a description. First, we formulate the exact meaning of using an LLM to "deduce" a fact from a paragraph. The process simulates human text comprehension, where the reasoner is impersonated by the LLM (Vicuna in our experiments). Having a change description and a fact, Vicuna is prompted with: "Here is a a description of some changes: *[change description]*. In the description, are there references to the fact that [**fact**]?" We further instruct the model to give a short answer, such that the LLM outputs "yes" or "no" at the beginning of the answer. A fact is a basic unit of true information, such as "A building has been removed in the center of

the scene", or "A new road appears in between houses".

The answer is analyzed using a simple word-matching rule to assign an affirmative or negative outcome. We found in all our experiments that using our prompt, the answers always start with "yes" or "no" and are followed by a brief explanation of the choice. We remove sampling during generation to obtain deterministic responses and avoid run biases. From now on, we refer to a fact as "covered" by a paragraph if it is deducible from it.

We then reason on the hypothesis that given a series of ground truth facts, the best description is the one that covers the majority of those facts. To evaluate a paragraph's *specificity*, all possible facts should be tested. This can be feasible or unfeasible, depending on the scenario. For instance, in the SECOND dataset, given the finite number of possible classes of change, we can define a finite set of facts. In contrast, with the LEVIR-CC dataset, formulating negative facts (changes that did not occur) is impossible. Therefore, we take only the 5 captions as *positive* facts.

Once we have a list of facts for each image $\mathbf{X_i}$, we can build two vectors, $\mathbf{p_i} \in R^J$ and $\mathbf{g_i} \in R^J$, where J is the number of facts for image i. $\mathbf{g_i}$ is the true matrix, containing the real outcome of each fact, while $\mathbf{p_i}$ is the model's predicted outcome for each fact. To populate $\mathbf{p_i}$, we take the generated change description for image i and use Vicuna to test each fact $Fj \in [1, J]$. We prompt the model as previously described and evaluate the answer of the model, which can be positive ("yes") or negative ("no"). We test each fact Fj and fill the prediction vector $\mathbf{p_{i,j}}$ with $+1$ if the LLM assigns a positive outcome, otherwise -1. The ground truth vector $\mathbf{g_{i,j}}$ is filled accordingly using the ground truth positive $(+1)$ or negative (-1) facts. For each fact f, the evaluation can raise three situations:

- **True Positive (TP)**: F_j is positive and the LLM test receives positive outcome ($\mathbf{g_{i,j}} = 1$ and $\mathbf{p_{i,j}} = 1$).
- **False Positive (FP)**: F_j is negative, and the LLM test receives a positive outcome ($\mathbf{g_{i,j}} = -1$ and $\mathbf{p_{i,j}} = 1$).
- **False Negative (FN)**: F_j is positive, and the LLM test receives a negative outcome ($\mathbf{g_{i,j}} = 1$ and $\mathbf{p_{i,j}} = -1$).

We compute sample-level true positive (TP_i), false positive (FP_i), and false negative (FN_i) by aggregating the results for an image i. For each image i, we can compute *precision* as $pr_i = \frac{TP_i}{TP_i+FP_i}$, *recall* as $rec_i = \frac{TP_i}{TP_i+FN_i}$ and F1 score $F1_i = \frac{2pr_i rec_i}{pr_i+rec_i}$. Scores for each image i are averaged over the entire test set to provide the final FMScore using Equation 6.1.

$$\text{FMScore} = \frac{1}{N}\sum_{i=1}^{N} F1_i \tag{6.1}$$

where N is the total number of images in the test set. The two extremes are:

- **1**: The ideal scenario. $\mathbf{g_i}$ and $\mathbf{p_i}$ are always identical. Only the true facts can be deduced from the descriptions.
- **0**: The worst scenario. $\mathbf{g_i}$ is the reciprocal of $\mathbf{p_i}$. $\mathbf{g_i} = \overline{\mathbf{p_i}}$. From the description, the opposite of each true fact can be deduced.

Generally, a better paragraph should correlate with an increase in FMScore.

6.5.1 Evaluating SECOND dataset

The SECOND dataset comprises semantic change maps for $K = 6$ classes of objects, yielding $F = K \times (K-1) = 30$ possible class changes for each couple of images. We excluded no-change scenarios,

where the start and end remain in the same class. We build $F = 30$ facts using the following template: "a *[class 1]* area has transformed into a *[class 2]* area", where *[class 1]* and *[class 2]* are placeholders for class names. Suppose the j-th element refers to a change from class w to class z. To populate $\mathbf{g_{i,j}}$, we analyze semantic change maps and assign a value of $+1$ if this change actually occurred somewhere inside the image, otherwise -1.

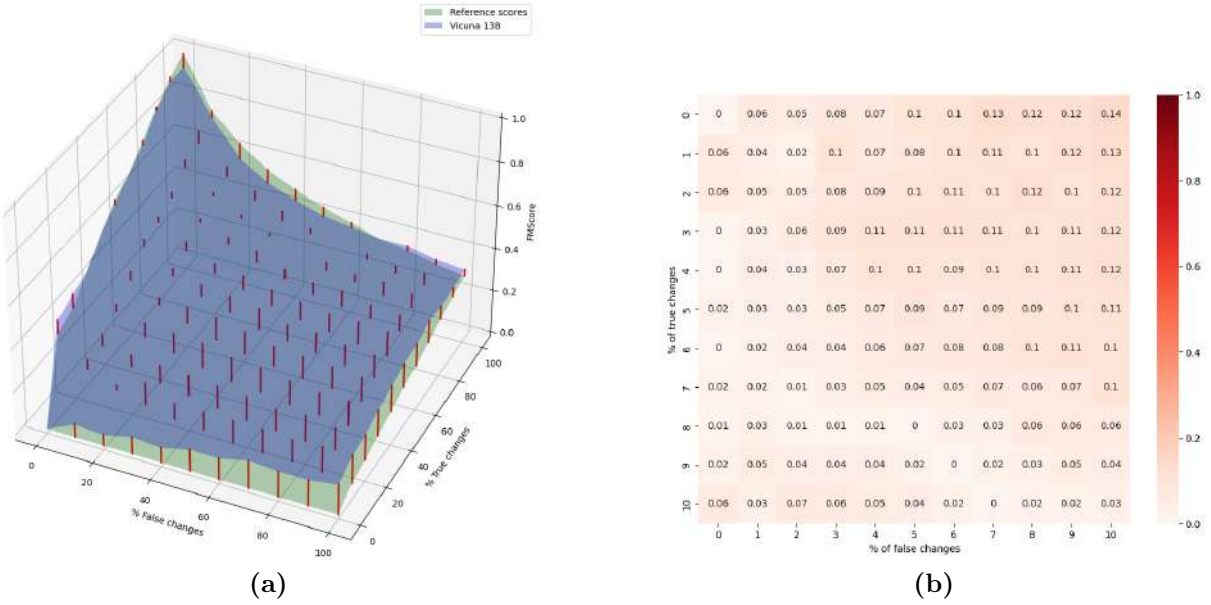
6.5.2 Evaluating LEVIR-CC dataset

The LEVIR-CC dataset includes ground truth captions detailing the changes, with each caption serving as an individual fact, amounting to 5 facts per image.

We populate each $\mathbf{g_i}$ with $+1$ since captions are regarded as changes that occurred between the two images. Consequently, the precision metric can only be $+1$ or 0, as the count of false positives is invariably 0.

6.5.3 Score sensitivity to the LLM

FMScore is based on the hypothesis that LLMs can coherently and effectively deduce facts from descriptions. Despite recent advances, with models that excel and sometimes surpass human capabilities in tasks such as text understanding and paraphrasing, we need to validate this hypothesis to ensure the reliability and applicability of the metric. In the following, we build a validation pipeline adopting a template-based approach using the SECOND dataset. For each image i, each of the 30 facts (class changes) is termed



(a) **(b)**

Figure 6.6: Validation results for Vicuna-13b-v1.5. (a) Validation of Vicuna for different percentages of true and false changes. (b) Absolute difference between reference f1-scores and Vicuna-derived f1-scores for different quality template descriptions. The lower, the better.

positive if it happened in image i; otherwise, it is negative. By concatenating different percentages of positive and negative facts, we can control the quality of the template description. We can generate the ideal description (100% positives, 0% negatives) and the worst description (100% negatives, 0% positives).

We generate descriptions of intermediate quality, varying the inclusion of positive and negative facts by increments of 10%.

We can thus compute the true reference score for each description (indicating the quality) and evaluate the alignment of our FMScore with this reference. An FMScore always equal to the reference indicates that the LLM coherently deduces which facts are described in the text. The results of Vicuna-13b-v1.5 on this task are plotted in Figure 6.6-a. We can see how the reference and our FMScore vary smoothly and consistently with paragraph quality. On the extremes, we got an FMScore of 0.94 for the perfect description (reference score 1) and 0.14 for the worst description (reference score 0). Figure 6.6-b depicts the absolute difference between the reference and FMScore across all combinations. Ideally, perfect alignment with the reference would yield a difference of zero. However, the average difference of 0.06 indicates that while the LLM's deduction process is generally acceptable, it is not flawless. This discrepancy might be attributed to the semantic correlation between specific changes. For instance, changes from "ground" to "building" and from "low vegetation" to "building" might lead to errors in the deduction process due to their semantic similarity.

The smooth variation of the FMScore supports the hypothesis that higher-quality paragraphs correlate with higher results. It is important to note that template descriptions could overestimate the metric's performance, as templates are simpler than real descriptions, making deductions more straightforward. Addressing this limitation is left for future work. Despite this, our analysis validates the hypothesis that Vicuna-13b-v1.5 can be used to derive our FMScore.

6.6 Results and discussion

In this section, we evaluate and compare the descriptions generated by our three proposed approaches. We decided to also compare our results to change descriptions obtained using GPT4-v, which is currently one of the best multi-modal models, although not free.

Table 6.1: Standard metrics scores of descriptions generated with the different approaches on SECOND and LEVIR-CC. BLEU (B1-4) $\in [0, 1]$, METEOR $\in [0, 1]$, CIDEr $\in [0, 10]$, ROUGE $\in [0, 1]$. Bold entries represent the best results.

Dataset	Approach	B1	B2	B3	B4	METEOR	CIDEr	ROUGE
SECOND	GPT-4	0.24	0.10	0.04	0.02	0.10	**0.01**	0.18
	Otter-direct	0.13	0.06	0.03	0.02	0.06	**0.01**	0.15
	Otter-indirect	0.13	0.05	0.02	0.01	0.06	**0.01**	0.19
	Otter-chat-open	**0.28**	**0.12**	**0.05**	**0.03**	**0.13**	**0.01**	**0.22**
	Otter-chat-template	**0.28**	**0.12**	**0.05**	**0.03**	**0.13**	**0.01**	**0.22**
LEVIR-CC	GPT-4	0.13	0.07	**0.03**	**0.01**	**0.12**	0.14	0.00
	Otter-direct	**0.19**	**0.08**	0.02	0.01	0.09	**0.20**	**0.01**
	Otter-indirect	0.11	0.05	0.01	0.00	0.08	0.13	0.00
	Otter-chat-open	0.08	0.04	0.01	0.00	0.09	0.08	0.00
	Otter-chat-template	0.09	0.04	0.01	0.00	0.09	0.09	0.00

Table 6.1 presents standard metrics results for the SECOND and LEVIR-CC datasets. The scores are generally low, indicating that the descriptions rarely match the exact wording of the ground truths. However, this does not necessarily imply lower paragraph quality, especially for longer and more complex outputs, as discussed in Section 6.4. On the SECOND dataset, the best results are obtained by Otter-chat-open and Otter-chat-template, while on LEVIR-CC the best results are obtained by GPT4-v and

Table 6.2: FMScore of descriptions generated with different approaches on both datasets. Bold entries represent the best results.

Dataset	Approach	FMScore	
SECOND	GPT-4	**0.46**	
	Otter-direct	0.09	
	Otter-indirect	0.07	
	Otter-chat-open	0.33	
	Otter-chat-template	0.35	
		All	Only Change
LEVIR-CC	GPT-4	**0.42**	**0.22**
	Otter-direct	0.16	0.00
	Otter-indirect	0.20	0.01
	Otter-chat-open	0.18	0.04
	Otter-chat-template	0.16	0.03

Otter-direct. An explanation of this deviation can be that on SECOND, obtaining the final description by summarizing the dialogue mimics the way in which the synthetic ground truths have been derived, resulting in more similar outputs.

Table 6.2 breaks down our FMScore for each approach and dataset. On LEVIR-CC, the results are further dissected considering two sets: all images and only those that exhibit some changes. GPT4-v achieves the highest scores, with 0.46 on SECOND and 0.42/0.22 on LEVIR-CC. Otter-chat-template and Otter-chat-open deliver competitive results on SECOND, while less competitive results on LEVIR-CC. Otter-direct and Otter-indirect yield lower scores on SECOND, indicating that the pre-trained Otter model is ineffective in directly extracting a change description on remote sensing images. Interestingly, these two approaches demonstrate better overall performance on the LEVIR-CC dataset. However, when focusing specifically on the subset of images that contain actual changes, a significant performance drop is observed. This discrepancy may arise from descriptions that are unrelated to the image content, leading the LLM to assign negative outcomes to the majority of facts. Consequently, since half of the images in LEVIR-CC are labeled as depicting no changes, this could result in an inflated FMScore. Observing the discrepancy in the results of the approaches using dialogue versus the approaches that solve the task directly, it seems that decomposing this complex task into smaller, more manageable tasks (answering precise questions) can facilitate the collection of coherent information. Some qualitative results are shown in Figure 6.11.

6.6.1 Class-wise FMScore breakdown

Figure 6.7 reports a class-wise FMScore breakdown on the SECOND dataset. We assign a change to a class if it comprises a transition either *to* or *from* that particular class. The results indicate that performance is generally lower for changes associated with *water* and *tree* classes. These classes are particularly challenging to detect in the SECOND dataset due to their representation as small pixel patches in the images (refer to Figure 6.8), making them susceptible to being obscured by variations in lighting conditions, seasonality, and other factors. Notably, GPT4-v achieves the highest scores across all classes. Conversely, Otter-chat and Otter-chat-template demonstrate strong performance in detecting changes related to *building* and *ground* classes but show reduced effectiveness on *low vegetation* and *sports field*. Both Otter-direct and Otter-indirect exhibit uniformly low performance across all classes.

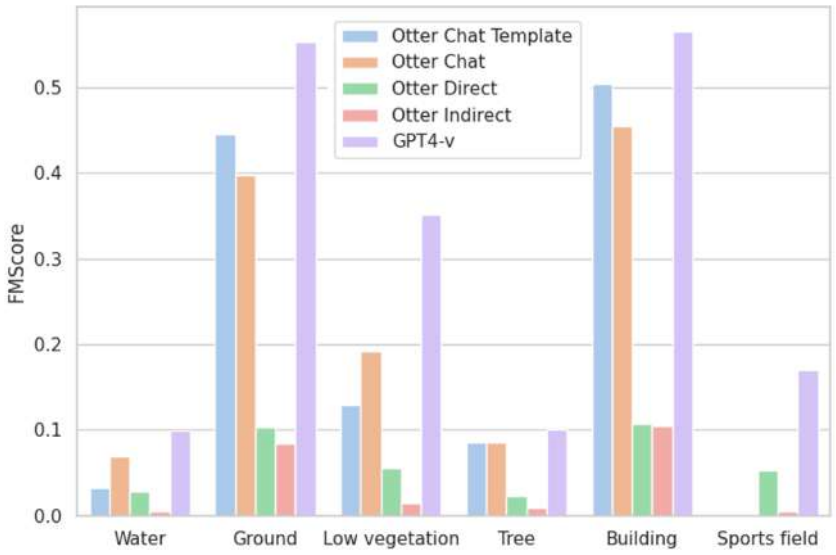


Figure 6.7: Class-wise FMScore for different approaches.

6.6.2 Effect of different template question sets

In this experiment, we vary the number of questions in the template set and evaluate the resulting description. We compared the original template set, with a reduced set containing fewer and an increased set with additional questions. The template sets can be found in Table 6.3 We removed or added batches of questions covering different concepts to better simulate real-world use cases. The template sets are reported in the appendix at the end of the chapter. The original template includes questions exploring changes in buildings and sports fields. The reduced set covers only questions exploring changes in buildings. The increased set covers questions about buildings, sports fields, and water areas. Results are depicted in Table 6.4. We observe that the FMScore varies with different templates. On LEVIR-CC, the score correlates with the number of questions in the template: the template with the highest number of questions achieves the highest FMScore. Conversely, the reduced set achieves the best score on the SECOND dataset.

Given our knowledge of the possible classes of changes on SECOND, we conducted another experiment to dissect class-wise performance using each template set. The results were more distinct and are depicted in Figure 6.9. For Example, in the water class, the extended set of questions showed greater performance due to the inclusion of questions exploring that concept. In contrast, the reduced set of questions failed to pick changes in the sports field class, primarily because it lacked questions related to that concept. The other classes exhibited trends consistent with the questions in the template sets, validating that defining a set of template questions can effectively direct the exploration of specific concepts.

6.6.3 Effect of automatic guidance prompts in Otter-chat

To further evaluate Vicuna's capability to autonomously generate questions pertinent to user interests, we conducted a test by explicitly guiding the LLM in formulating questions. This approach aims to alleviate the manual effort required to create template questions. By incorporating an "Intention" prompt (Table 6.6) alongside the "First question" prompt in Otter-chat, we can guide the LLM in producing questions. We explored two scenarios, simulating two different user intentions: the first directs the

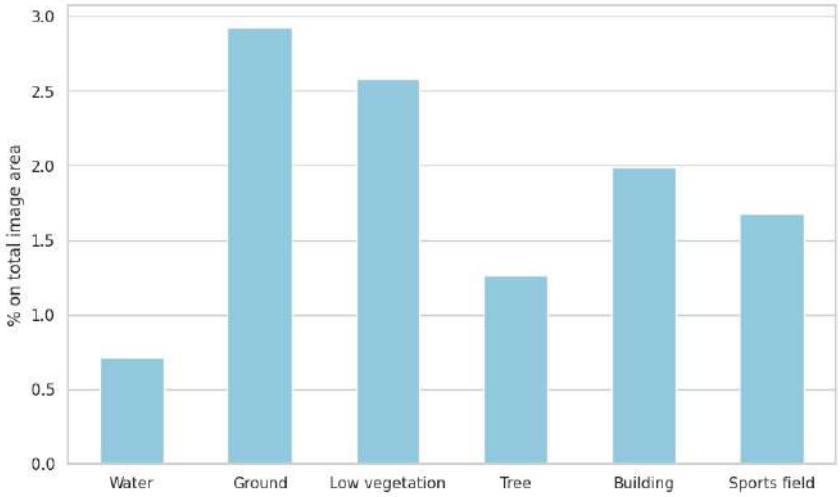


Figure 6.8: Average % of image area covered by class-wise changes.

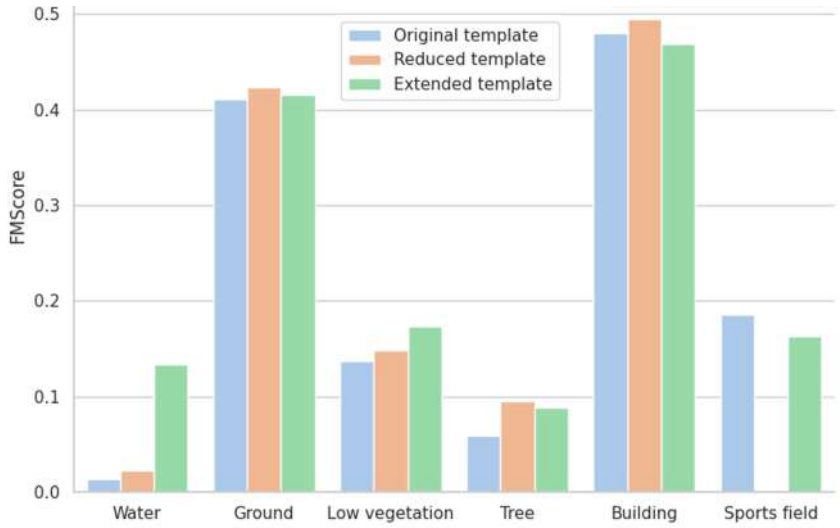


Figure 6.9: Class-wise FMScore of Otter-chat-template using different sets of template questions on the SECOND dataset.

model to focus on changes related to buildings, while the second on changes related to buildings *and* sports fields. The outcomes, presented in Table 6.5, indicate that on SECOND, the highest performance was achieved without additional guidance. Conversely, on LEVIR-CC, the best results were obtained with explicit guidance towards changes concerning buildings. A detailed class-wise performance analysis, depicted in Figure 6.10, reveals the impact of guidance on specific classes. Notably, the prompt directing Vicuna to generate questions about buildings and sports fields yielded the highest score for the class "sports fields," demonstrating successful adherence to the guidance. Regarding the class "water," the performance declined under explicit guidance towards buildings and sports fields, indicating a successfully narrowed focus on the specified classes.

Table 6.3: Comparison of original, reduced, and increased checklists.

Original	**Reduced**	**Increased**
- Have there been any changes in the appearance of buildings between the two images? - Are there any signs of damage to the buildings between the two images? - Have new buildings been constructed in the area? - Have buildings been removed between the two images? - Have parts been added to the existing buildings between the two images? - Are there signs of new construction sites in the area? - Have new playgrounds appeared in the area? - Have there been any changes in the appearance of playgrounds between the two images? - Have playgrounds been removed between the two images? - Are there any signs of damage to the playgrounds between the two images?	- Have there been any changes in the appearance of buildings between the two images? - Are there any signs of damage to the buildings between the two images? - Have new buildings been constructed in the area? - Have buildings been removed between the two images? - Have parts been added to the existing buildings between the two images?	- Have there been any changes in the appearance of buildings between the two images? - Are there any signs of damage to the buildings between the two images? - Have new buildings been constructed in the area? - Have buildings been removed between the two images? - Have parts been added to the existing buildings between the two images? - Are there signs of new construction sites in the area? - Have new playgrounds appeared in the area? - Have there been any changes in the appearance of playgrounds between the two images? - Have playgrounds been removed between the two images? - Are there any signs of damage to the playgrounds between the two images? - Have new water bodies appeared in the area? - If there are water bodies, have they changed in size between the two images? - Have water bodies disappeared from the first to the second image?

These findings validate the effectiveness of using a broad guidance prompt to steer Vicuna, thus significantly reducing the burden of creating template questions. This method allows potential users to provide general instructions, enabling the LLM to autonomously generate relevant questions.

6.6.4 Visual results of LLM's ability to "deduce" facts

Another significant aspect of our analysis is the evaluation of LLMs' ability to detect the presence of facts within paragraphs. We expanded our analysis by including visual examples of this ability, incorporating two examples using GPT-3.5 as the LLM for fact deduction. Visual results are illustrated in Figures 6.12 and 6.13. The figures include a matrix on the left side, which depicts the ground truth changes between

Table 6.4: FMScore obtained using different sets of template questions. Higher is better.

Dataset	**FMScore**		
	Original	Reduced	Extended
SECOND	0.32	**0.34**	0.32
	Original	Reduced	Extended
LEVIR-CC	0.16	0.13	**0.18**

Table 6.5: FMScore obtained using different guiding prompts in Otter-chat-open. Bold entries represent the best results.

Dataset	**Guiding Prompt**	**FMScore**
SECOND	-	**0.33**
	I am only interested in changes related to buildings.	0.31
	I am only interested in changes related to buildings and sports fields	0.27
LEVIR-CC	-	0.18
	I am only interested in changes related to buildings.	**0.21**
	I am only interested in changes related to buildings and sports fields	0.20

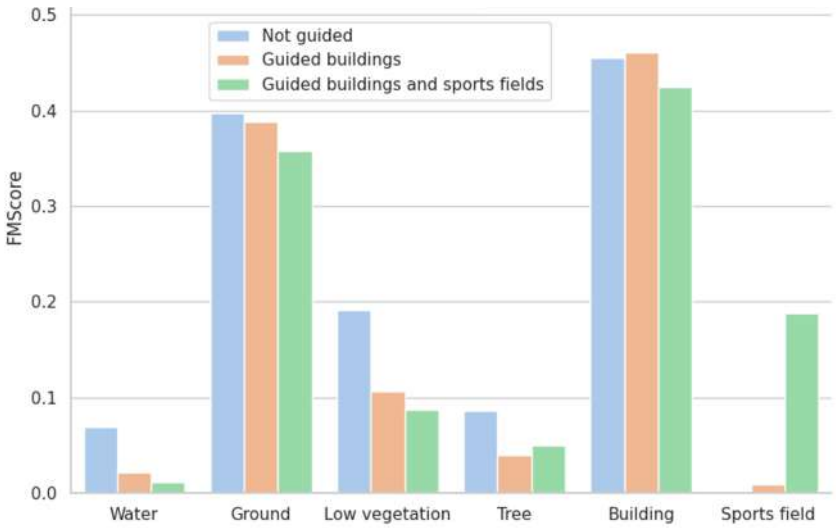


Figure 6.10: Class-wise FMScore of Otter-chat using different Intention Prompts on the SECOND dataset.

two images. In this matrix, yellow squares indicate that somewhere in the image, there is a transition from the class on the x axis to the class on the y axis (positive fact). Violet squares signify the absence of such change (negative fact). Each row represents a class in image $\mathbf{X}_1$, and each column corresponds to a class in image $\mathbf{X}_2$. For instance, a yellow entry at position $(1, 4)$ indicates that the fact "a ground area has transformed into a building area." is positive for that image. As a further example, in Figure 6.12, the following changes are observed between the images:

- A ground area has transformed into a low vegetation area.
- A ground area has transformed into a building area.
- A low vegetation area has transformed into a ground area.
- A low vegetation area has transformed into a building area.

In our analysis, we tested the deduction of each fact from the generated paragraph using both Vicuna and GPT-3.5 and calculated their respective FMScore. For instance, the description generated by GPT-4v, as illustrated in Figure 6.12, received from Vicuna a positive outcome for all and only the true facts, resulting in an FMScore score of 1. A manual inspection of the text reveals that most decisions are well-supported by the description, except for the change from *ground* to *low vegetation*. The evaluation using GPT-3.5 indicates a more conservative behavior. The model detects fewer changes yet demonstrates greater precision. Specifically, all the changes identified by the model are supported by the description. The sentence "some vegetation appears to have grown significantly between the two images" highlights a key difference in interpretation between Vicuna and GPT-3.5. Vicuna interprets this sentence as indicating a change from *ground* to *low vegetation*, while GPT-3.5 does not recognize this change. Despite the reasonableness of assigning the presence of this class of change based on the sentence, GPT-3.5 adopts a stricter evaluative approach, requiring more explicit statements to confirm the presence of facts. Nonetheless, GPT-3.5 is not flawless, as evidenced by its incorrect identification of a change from *ground* to *sports field* in the text generated by Otter-direct, as shown in Figure 6.13.

6.7 Conclusion

This chapter proposes, develops, and analyzes three novel approaches for generating remote sensing change descriptions using pre-trained large language and multi-modal models (LLMs and VLMs). We provide a thorough motivation for our methodological choices, detailing the strengths and weaknesses of each approach. Additionally, we introduce and validate a new evaluation criterion called *Fact Matching Score (FMScore)*. This criterion leverages a large language model to deduce the presence or absence of references to factual information inside a description, framing its coherence as how many real "facts" can be deduced from its contents. We report quantitative and qualitative results of our three approaches on two remote sensing datasets: SECOND and LEVIR-CC. We expanded our evaluation by including GPT4-v generated change descriptions as a comparison. Our findings demonstrate that while GPT-4V sets a strong baseline, our proposed approaches achieve competitive results. Notably, our methods are open source and can be seamlessly integrated with more specialized and robust large language and multi-modal models to enhance performance.

Table 6.6: Prompts used in our approaches

Approach	**Subtask**	**Model**	**Prompt**
Direct	-	Otter	Describe the changes between these two views of the same area at different times. Describe only the changes, if there are no noticeable changes, mention it accordingly.
Indirect	Description	Otter	Describe in detail this satellite image.
	Extraction	Vicuna	I have two descriptions of two satellite images. Based on the two descriptions, create a paragraph summarizing the main changes that you deduce. Do not describe what is in each image but just changes. If there are no significant changes, just say that. Description 1: - text-. Description 2: -text-. Changes:
Chat-open	First question	Vicuna	I am interested in knowing which changes happened between two satellite images of the same area acquired at different times. I can look at the two images and answer your questions one by one. You must produce questions focused on structural changes in the area layout. It must be possible to answer the questions just by looking at the rgb images. First question:
	Intention	Vicuna	I am only interested in changes related to ...
	Next question	Vicuna	Next question:
	Answer	Otter	Answer the question. If you are unsure on the answer, say you don't know.
	Summarization	Vicuna	This dialogue is about possible changes between two satellite images. Summarize it creating a descriptive paragraph of the changes happened. The paragraph should not mention that the information is taken from the dialogue, it must be just a description of the changes as it would be by looking at the images.
Chat-template	Summarization	Vicuna	This dialogue is about possible changes between two satellite images. Summarize it creating a descriptive paragraph of the changes happened. The paragraph should not mention that the information is taken from the dialogue, it must be just a description of the changes as it would be by looking at the images.
GPT4-v	-	GPT4-v	I'm a user interested in textual descriptions of the changes occurred between two satellite images and you are an assistant that can help me in this task. I will upload both images together in the prompt. You must directly describe the changes between them without relying on additional tools. I'm interested in structural changes (for Example the construction of a new building, changes in land use, and others that are pertinent to remote sensing). You must avoid to describe changes that are related to the image acquisition or seasonality.

OTTER INDIRECT
There are no significant changes between the two satellite images. Both images show a residential area, with houses and possibly a fence. However, while the first image has a person standing next to a fence in the foreground, the second image has several cars parked along the street in the foreground.

OTTER CHAT TEMPLATE
Looking at the two satellite images side-by-side, it is evident that there have been several changes in the area between the two dates. In the later image, there are more buildings and structures visible compared to the earlier image. Additionally, there is a new building visible in the right half of the image, which was not present in the earlier image. Moreover, the parking lot and road layout seem to have changed slightly. Another noticeable difference is the presence of a small building in the left half of the image, which was not present before. Lastly, there seems to be a new residential complex in the center of the image, which was also absent in the earlier image.

OTTER DIRECT
The changes between the two views are minimal and mostly confined to the edges of the image. In the first image, there are several parked cars along the street, but no cars are moving. In contrast, the second image features cars moving on the street; it seems to be a different time of day or a different day altogether.

OTTER CHAT
Upon comparing the two satellite images, several notable changes can be observed in the area. In the second image, new buildings have been constructed, including a large structure that appears to be a shopping center. Additionally, a new road has been built, connecting two previously separate sections of the city. Several new parking lots and garages have also been added, suggesting increased commercial activity in the area. Finally, there appear to be new trees and landscaping throughout the area, likely contributing to a more visually appealing environment. Overall, these changes suggest a growing and developing community.

GPT4-v
Looking at the second image, a significant urban development is observed where a large area of vegetation has been removed to make way for residential or commercial buildings, along with the construction of these new structures, additional roads have been established, carving through what was once a green space to create a structured urban grid. this development has led to an increase in the density of buildings, expanding the urban area and creating a defined boundary where before it was more irregular and interspersed with natural vegetation.

OTTER INDIRECT
The two satellite images show different locations. Image 1 shows a city street lined with houses and cars, while Image 2 shows a close-up view of a building with multiple rooms, possibly a house or a small apartment building. There are also more people and objects present in Image 2, including a bench in the middle of the scene.

OTTER CHAT TEMPLATE
Upon comparing the two satellite images, several notable changes can be observed. In the second image, a new building seems to have been constructed near the waterfront. Additionally, what appears to be a parking garage has been built in the same location. On the other hand, some buildings that were present in the first image are no longer visible in the second image, suggesting they may have been demolished or removed. Furthermore, there is a noticeable change in the appearance of the waterfront itself, which now features a wider promenade and possibly renovated seating areas. Lastly, while the playgrounds remain in the same location, there are indications of new construction nearby in both images. Overall, the area appears to have undergone significant development and infrastructure changes.

OTTER DIRECT
In the first image, there is a bridge crossing a river near a forest. In the second image, the same bridge is still visible, but the surroundings have changed. There is now a road near the river, and the forest has been cleared. This indicates that there might have been development or changes in the area in the time between the two images.

OTTER CHAT
The two satellite images show a residential area, with a sidewalk and streetlight pole visible in both pictures. While the general layout of the area remains consistent, there are several subtle changes between the two images. The most noticeable difference is the absence of a wooden bench in the second image. Additionally, the position of the streetlight pole seems to remain unchanged. In terms of the environment, the trees in the background appear to be the same in both images, although their exact appearance is not specified. Finally, while the first image captures a car passing behind the camera, the second image does not seem to show any cars in the vicinity, indicating a potential change in vehicle traffic or parking habits.

GPT4-v
The most striking change as the area transitions from the first to the second image is the realignment of the railway tracks, which now take a more direct, straighter path compared to the previous curved route. This alteration likely enhances the efficiency of the transportation network. Adjacent to these tracks, there has been an expansion of the industrial facilities, indicated by the construction activity in progress. The completion of a large blue-roofed building at the bottom enhances the industrial skyline. Furthermore, the green belt along the top of the image has thickened, suggesting either natural growth over time or deliberate efforts to bolster the area's vegetation.

OTTER INDIRECT
There are no significant changes between the two satellite images described. Both images depict a rooftop garden.

OTTER CHAT TEMPLATE
Between the two satellite images, several changes can be observed. In the second image, a new building can be seen, while the building on the left side of the first image is missing. Additionally, the sports field appears to have been removed. Some vegetation growth can also be noticed near the building in the second image. Finally, a new construction site can be seen in the second image.

OTTER DIRECT
The angle, or posture, of a photograph taken at X: 223 Y: 167 is different from the second photograph, indicating a change.

OTTER CHAT
Upon comparing the two satellite images, several notable changes can be observed in the area. Firstly, a new building has been constructed in the center of the images, which was not present in the earlier image. Additionally, the parking lot adjacent to the new building has also been expanded. Furthermore, some of the trees around the perimeter of the park have been cut down, while others have grown significantly taller. Finally, there have been some changes to the road network in the area, including the addition of a new roundabout near the park entrance.

GPT4-v
In the second image, a well-developed residential area appears where there was previously an undeveloped, barren land in the first image. Multiple houses have been constructed, complete with roadways that define the street layouts. Driveways and parking spaces are also visible, along with several vehicles parked on the newly constructed roads. The transformation from an empty plot to a developed residential neighborhood is prominent.

OTTER INDIRECT
No significant changes can be deduced from the descriptions as both descriptions indicate that the same rooftop garden is being described, which remains consistent over time.

OTTER DIRECT
I backlit an outdoor greenhouse and captured a snapshot of it during a daytime scene.

GPT4-v
In the second image, a significant transformation can be observed. A new, wider road or path has been constructed, taking over one of the less-defined tracks present in the first image. Additionally, a building or structure has been added in the bottom right corner, which was not present in the initial image.

OTTER CHAT TEMPLATE
Upon comparing the two satellite images, several changes can be observed. In the second image, there is a newly built structure located at the edge of the parking lot, while some trees have also been planted. Additionally, the baseball diamond appears to have been reconstructed with new dugouts and a new infield. The outfield fence has also been realigned further away from home plate. However, it appears that the tennis courts have been removed and replaced with what looks like a new multi-use court. The track surrounding the football/soccer field has also been repaved and the field itself has been resurfaced. Furthermore, a new access road has been constructed, which now leads to a new building that was not present before.

OTTER CHAT
Between the two satellite images, several changes can be observed in the area. The most significant change is the construction of a new building in the center of the image. Additionally, the parking lot near the building has been expanded, and the surrounding vegetation appears to have grown significantly. There are also some minor changes in the road network, including the addition of a roundabout and modifications to the road leading to the building. Lastly, there is a noticeable shift in the position of the sun, indicating that the images were taken at different times of the day.

Figure 6.11: Qualitative results on SECOND (top) and LEVIR-CC (bottom).

Figure 6.12: Manual verification of fact presence evaluation by Vicuna 13B (top) and GPT-3.5 (bottom). Green: correct passages. Red: wrong passages. Yellow: borderline passages. In matrices, we manually inspect the paragraphs and mark with green, yellow, and red the correct, halfway correct, or wrong deduction of each (fact), respectively.

Figure 6.13: Manual verification of fact presence evaluation by Vicuna 13B (top) and GPT-3.5 (bottom). Green: correct passages. Red: wrong passages. Yellow: borderline passages. In matrices, we manually inspect the paragraphs and mark with green, yellow, and red the correct, halfway correct, or wrong deduction of each (fact), respectively.

Chapter 7

Exploring the use of Ancillary Geographic Information to Enrich Captions

Based on work in progress.

7.1 Introduction

In Chapter 3, we investigated visual dialogue to enrich the descriptiveness of remote-sensing image descriptions. The idea was to mine for further information that can be deduced by looking at the image contents. This paradigm can incorporate general information that is not specific to the current scene but is broadly applicable to any scene with similar visual characteristics. In this context, we think that a useful property lies in the ability of a captioning model, or a visual assistant in general, to add specific information that is valid *only* for the scene we are looking at. For example, wouldn't it be useful to have a system that can report the name and purpose of a building in an image or direct a user to useful information regarding a monument inside a particular scene? This information is not inferrable if you only look at the image; something else is needed.

Geographic databases are geo-referenced spatial databases used for storing and manipulating information on geographic features (i.e., data associated with a location on Earth). One prominent example is OpenStreetMap (OSM), an open-source effort to build a comprehensive and complete world map. OSM contains information about many geographic features, such as buildings, roads, hospitals, schools, stadiums, etc. Merging the visual ability with the ability to understand information about geographic features could help build assistants that can answer with more coherence and more details about the contents of an image. The only relevant work we found in this area is [168]. The authors utilize OpenStreetMap data to create captions incorporating specific scene details based on the geographical context. These captions are brief and focus on elements in close proximity to the scene, which is captured from a ground-level perspective. Nonetheless, their work highlights the potential of integrating GIS data into remote sensing applications, generating captions that offer valuable insights into the image's location and nearby features.

On the same line, we want to test the possibility of using OSM data to provide more grounded detail for images captured from an aerial perspective. Furthermore, unlike the authors of [168], we want to add specific details only to geographic features inside the scene.

The primary focus is generating a dataset of triplets (image, OpenStreetMap data, description). To evaluate if the system can produce richer captions, two types of descriptions are required: one that provides general information and another that offers targeted details specific to the scene based on its geographic features. Secondly, data integration: developing a deep-learning model to process the image and additional geographic data for inference is complex. Moreover, images might contain varying geographic features in different quantities, requiring the model to adapt to these variations. This chapter explores initial approaches to integrating geographic information into a remote-sensing image captioning pipeline. Drawing inspiration from the extensive and freely accessible textual data in OpenStreetMap, which covers thousands of geographic features, we propose that a system capable of understanding and presenting this information in an integrated manner would enhance the user experience, providing more accurate and grounded responses.

7.2 OpenStreetMap data

The OpenStreetMap database is an open-source effort to map the world. In OSM, users can map thousands of features, from parks to lakes, ponds, streets, houses, museums, stadiums, etc. OpenStreetMap comprises three basic geographical features: nodes, ways, and relations. Nodes (also called point features) represent individual points in space defined by their latitude and longitude. For example, a park bench or a water well can be mapped using nodes. Ways are collections of nodes and can be either open or closed. Open ways generally represent roads or rivers, while close ways form areas representing parks, buildings, residential areas, etc. Finally, relations define complex relationships between multiple nodes, ways, and other relations. An example is the definition of a trail as the union of multiple roads (ways). Each geographic feature carries its latitude, longitude position, and a list of *textual tags*. The tags are used to define the properties of the feature.

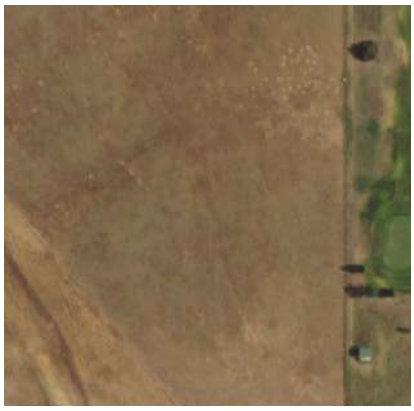
Examples of images with the corresponding geographic features from OSM are provided in Figure 7.1. Every feature represents an "object" inside the image. Tags are stored as *key:value* pairs. Since users have the flexibility to define custom keys and values, there is no strict regulation. However, over time, a standard has been established to regulate tagging. Today, a wiki detailing best tagging practices is available online, ensuring that similar objects (e.g., buildings) are typically tagged with the same root tag (e.g., building=yes). Looking at the example in Figure 7.1, we can see the tag "landuse:industrial". This likely represents the area on the bottom right, specifying that it is an industrial site. We can also see that OSM tags point to the presence of a railway, which is not readily identifiable by looking at the image. Also, from the example, we can see how different scenes can be defined by different amounts of additional OSM information, highlighting the need for a system that can robustly handle shifts in the amount and type of tags related to geographical features.

7.3 Dataset

We collected a dataset of 1568 images with additional data about geographic features within the image from OpenStreetMap. We restricted our focus to high-resolution images to reference and describe objects with high detail. We used images acquired from the United States of America for two main reasons. First, the NAIP[1] program provides free high-resolution remote sensing imagery over many USA countries, with a ground resolution of 0.6 meters/pixel. Second, the USA is well-mapped in OpenStreetMap, providing a fertile ground for our exploration.

[1] https://naip-usdaonline.hub.arcgis.com/

{crossing:barrier: no, railway: level_crossing, position:center right}
{building: yes, position: top right}
{building: yes, building:levels: 1, position: top right}
{landuse: industrial, position: top right}
{building: yes, position: bottom right}
{building: yes, position:bottom right}
{natural: water, water: pond, position: center}
{highway: unclassified, maxspeed 25 mph, maxweight: 10 st, surface: asphalt, position: from top right to topcenter}
{electrified: no, gauge: 1435, railway: rail, usage: branch, position: from top left to center right}
{highway: service, position: top center}
{highway: service, position:top center}
{highway: service, position: fromtop center to top right}



{'leisure': 'golf_course', 'name': 'Cimarron Golf Club', 'position': 'bottom right'}

Figure 7.1: Two images captured over the USA, paired with geographic feature tags sourced from OpenStreetMap. In the top image, a water feature in the center is labeled as a pond. The bottom image shows a golf course on the right, with the name in the tags. OSM data is crucial for identifying its presence for certain features, like the railway in the top image. Similarly, the golf course in the bottom image, which mostly lies outside the view, is not easily recognizable from the image alone.

7.3.1 Data Collection

To build our dataset, we proceeded in two steps: first, we identified a list of states from which to collect data, depicted in Table 7.1. The choice of states is arbitrary, and we tried to collect images from different states to avoid biases due to weather conditions. Secondly, we identify a list of geographic features of interest, depicted in Table 7.2, to limit scene variability and ensure that specific features are covered in the dataset. The data collection process is carried out identically for each USA state. First, we divide the large image tiles from NAIP into a uniform grid of cells of dimension 256×256 meters. Then, we randomly select cells and use their coordinates to interrogate OSM to retrieve the list of geographic features inside the scene. We verify if at least one geographic feature is included in the list of features of interest. If the answer is affirmative, we download the corresponding image and resize it to 512×512 pixels. We scrape information from OSM about all the features within the scene, restricting to "nodes" and "ways" geographic features. We decided to avoid "relations" features as they are more complex to model and leave their exploration as a future endeavor. With this step we collect 1568 images with corresponding OSM data, distributed according to Figure 7.2.

Table 7.1: List of States

No.	State	No.	State
1	Idaho	2	Georgia
3	Florida	4	Delaware
5	Connecticut	6	Colorado
7	Arkansas	8	Arizona
9	Alabama	10	Illinois
11	Indiana	12	Iowa
13	Kansas		

Table 7.2: List of geographical features of interest used to scrape images for our dataset.

Key	Value	Key	Value
building	stadium	leisure	golf_course
military	bunker	sport	baseball
amenity	place_of_worship	landuse	forest
highway	service	amenity	bank
aeroway	aerodrome	highway	trunk
natural	beach	building	bunker
amenity	parking	man_made	storage_tank
natural	desert	sport	basketball
building	house	natural	wood
landuse	residential	highway	primary
waterway	river	sport	golf
amenity	restaurant	highway	motorway
leisure	stadium	leisure	swimming_pool
service	parking_aisle	amenity	hospital
sport	tennis	leisure	park
building	yes	highway	residential
natural	water	landuse	farmland

7.3.2 Geographic location encoding

Each geographic feature inside OpenStreetMap is linked to its position on the earth, given by a (latitude, longitude) pair. Node features have only one pair of coordinates, while way features, which are collections of nodes, have a list of coordinates. We used a set of locations, depicted in Figure 7.3, referring to specific image regions to convert the raw coordinates into textual locations. In detail, we divide the image uniformly into nine regions, each assigned to a textual locution specifying its position inside the image. First, we map world coordinates into image coordinates ($(0, 0)$ is in the top left corner). For node features, the cell where the feature is located specifies its position locution. For closed ways, we consider the center of the area as the feature coordinate and use the same process as node features. For open ways (roads, rivers), we use the start and end node coordinates, describing the position of the feature using the sentence "from x to y," where x is the cell enclosing the start node, and y is the cell enclosing the end node. Figure 7.1 shows an example of our encoding procedure, where the tags contain our added key, specifying the position. All these transformations are made on the fly so that the original OSM data remains untouched and can be leveraged by researchers willing to test other methodologies to embed

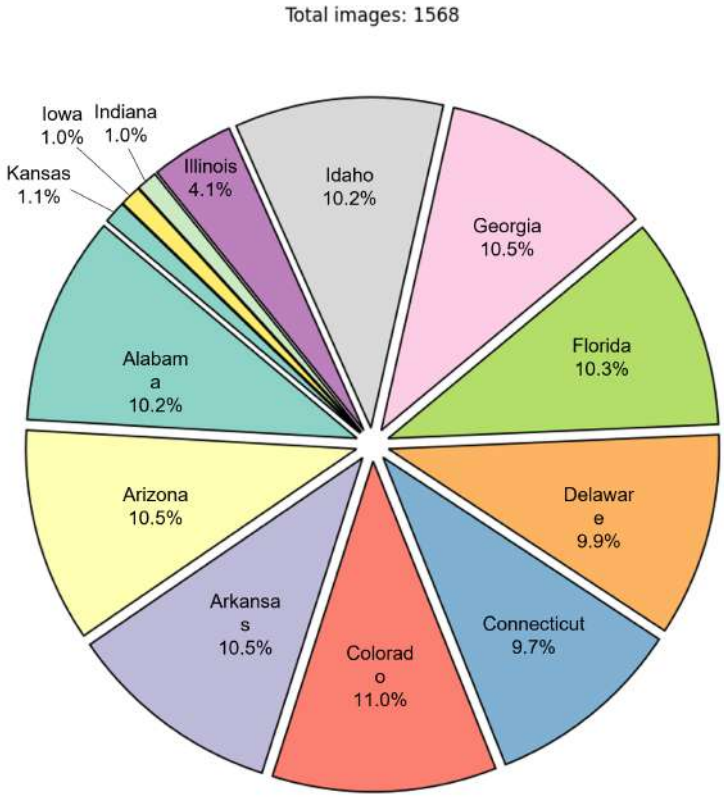


Figure 7.2: Image per state distribution of images in our dataset.

the position. Furthermore, we think more sophisticated integration schemes can leverage the detailed position information from geographic information systems to further increase the grounding of the visual information in such systems.

7.3.3 Labeling

We manually label each image with two separate descriptions. The first is a general description of the image content. The rationale is that the information in the general description must be visually inferrable by only looking at the image. When generating general annotations, the annotator has access to the additional OSM data to be more precise and consistent with the annotations, with the explicit request to avoid inserting details in the description that cannot be readily inferred from the image alone. When generating the augmented description, the annotator is instructed to insert additional information specific to the scene, incorporating particular details from the OSM data that cannot be inferred by the image alone. Some examples of samples in our dataset are depicted in Figure 7.4. Since the general and augmented descriptions of Figure 7.4 appear different, misleading the reader, we specify that as a data augmentation strategy, we create 9 additional versions of both descriptions using chatGPT. Specifically, we input the original description and instruct ChatGPT to create 9 alternative versions that preserve the original meaning while varying the wording. The same is done for augmented descriptions. This is

Figure 7.3: Grid to map feature position from world coordinates (latitude, longitude) to textual locations.

the reason for the different wording of the general and augmented descriptions. However, they keep the same semantic meaning as the original annotated by us and thus are comparable. In green, some details of the augmented description are specific to the scene and not inferrable from the visual content. With text data augmentation, our dataset includes 10 general descriptions and 10 augmented descriptions for each image.

7.3.4 Text statistics

Our manually labeled dataset has a higher degree of description richness. Our captions thoroughly describe the scene, the objects' position, appearance, and relations between them. Besides, the augmented descriptions also include information from OSM tags, including details that cannot be inferred from the image alone. As can be seen from the two examples of Figure 7.4, the captions describe each aspect of the image, including the conditions of the objects (multi-lane asphalted highway, golf course with manicured lawns and uneven edges), the network of roads, the relative positions between objects (flat-roofed building with a parking lot in front), and so on. In Table 7.3, we summarize some statistics of our dataset compared with other datasets in the literature. We can see that in our dataset, every image is associated with at least one caption that is entirely unique across the dataset. Additionally, thanks to the data augmentation step using chatGPT to rephrase the descriptions, 60.52% of the general descriptions contain unique words that do not appear in any other description. This percentage increases to 67.98%

General description
The image features a golf course with manicured lawns and uneven edges, surrounding a dark blue pond. Small paths provide access for players. In the top left corner, a section of a larger asphalt road is visible. Surrounding the course are sparse patches of bright green vegetation, low vegetation, and bare soil.

Augmented description
A golf course called Baywood Greens, with manicured lawns and uneven boundaries, surrounds a dark blue pond in the middle of the scene. Small cartpaths run around the course, allowing players to access it. A portion of a bigger asphalted road is visible in the top left corner. Surrounding the course are sparse patches of bright green vegetation, low vegetation, and bare soil.

General description
A developed area showcasing a network of roads and buildings. In the center, a multi-lane asphalted highway runs diagonally. To the left side of the image is a large flat-roofed building, likely a commercial or industrial facility, with a parking lot in front. On the right side, other structures, likely houses or smaller shops, are interleaved with areas of curated lawns.

Augmented description
The high-resolution remote sensing image depicts a developed area showcasing a network of roads and buildings. In the center, a major asphalted highway runs vertically, with multiple lanes visible. To the left side of the image, a large white building belongs to the New Castle County Airport. On the right, other structures, such as an alcohol shop and a gas station of BP, are interleaved with areas of curated lawns.

Figure 7.4: Examples of images with corresponding general and augmented descriptions. Green highlights details that only appear in OpenStreetMap data.

when analyzing the augmented version of the descriptions. This is primarily due to the inclusion of specific names of geographic features (names of parks, golf courses, airports, etc.), which are often unique to a single scene.

7.3.5 Key Filtering

As we saw previously, in OSM a tag is a key:value textual pair specifying some characteristic about a geographic feature. OSM contains 99 thousands different keys[2]. This huge variability can prove problematic for our scope, as it introduces the additional difficulty of handling different types of tags that can contaminate the input. To reduce this variability, we created a blacklist of allowed keys such that we can filter out unwanted tags and only keep the most interesting ones. Before key filtering, our dataset contains 315 different keys. After filtering, we reduce this pool to 66, depicted in Table 7.4. Key filtering is applied on the fly during training and inference, and it does not affect the original OSM tags in our dataset, so other researchers can experiment with other strategies.

[2]https://wiki.openstreetmap.org/wiki/Tags

Table 7.3: Statistics of some representative datasets for RS image captioning. General: general descriptions in our dataset. Augmented: augmented descriptions in our dataset.

	UCM [35]	RSICD [36]	VRSBench [169]	General	Augmented
N images	2100	10921	19,805	1,568	1,568
Vocabulary Size	368	3,325	8,387	7,295	7,168
Total Caps	10,500	54,605	19,805	15,680	15,680
Unique Caps	2,032	18,190	19,800	15,629	15,589
% Img w Unique Cap	14.90%	72.96%	99.95%	100.00%	100.00%
% Cap w Unique Words	1.52%	10.43%	12.70%	60.52%	67.98%

Table 7.4: The pool of retained keys.

Retained keys			
aerodrome:type	aeroway	amenity	attraction
bicycle_parking	building	building:colour	building:material
bunker_type	construction	content	country
crop	crossing	denomination	designation
footway	generator:method	generator:source	generator:type
golf	healthcare	highway	historic
industrial	intermittent	junction	landuse
leaf_cycle	leaf_type	leisure	man_made
material	memorial	military	military_service
natural	operator:type	ownership	parking
parking_space	pipeline	place	plant:method
plant:source	playground	position	power
railway	religion	service	shelter_type
shop	social_facility	sport	substance
substation	support	surface	swimming_pool
tourism	tower:type	water	waterway
wetland	wholesale	alt_name	brand
county	cuisine	name	official_name
old_name	operator		

7.4 Preliminary Exploration

7.4.1 Architecture

To analyze the potential to generate enriched captions with additional OSM tag information, we adopted an encoder-decoder architecture, where a vision transformer image encoder maps an image $\mathbf{X}$ to a sequence of visual embeddings Z_x, and a language model decoder generates the target caption conditioned on the visual information. We designed a dedicated module to accommodate the additional textual data from OpenStreetMap, treating OSM tags as pure textual metadata. Figure 7.5 depicts the architecture of our model. In detail, the OSM tags are first filtered with the blacklist. Then, the key-value pairs of the remaining tags are concatenated in a long textual string, which is tokenized and embedded using the same vocabulary used for the caption. Furthermore, an object embedding is added to the tokens of each geographic feature to relate tags with the corresponding geographic features in the attention

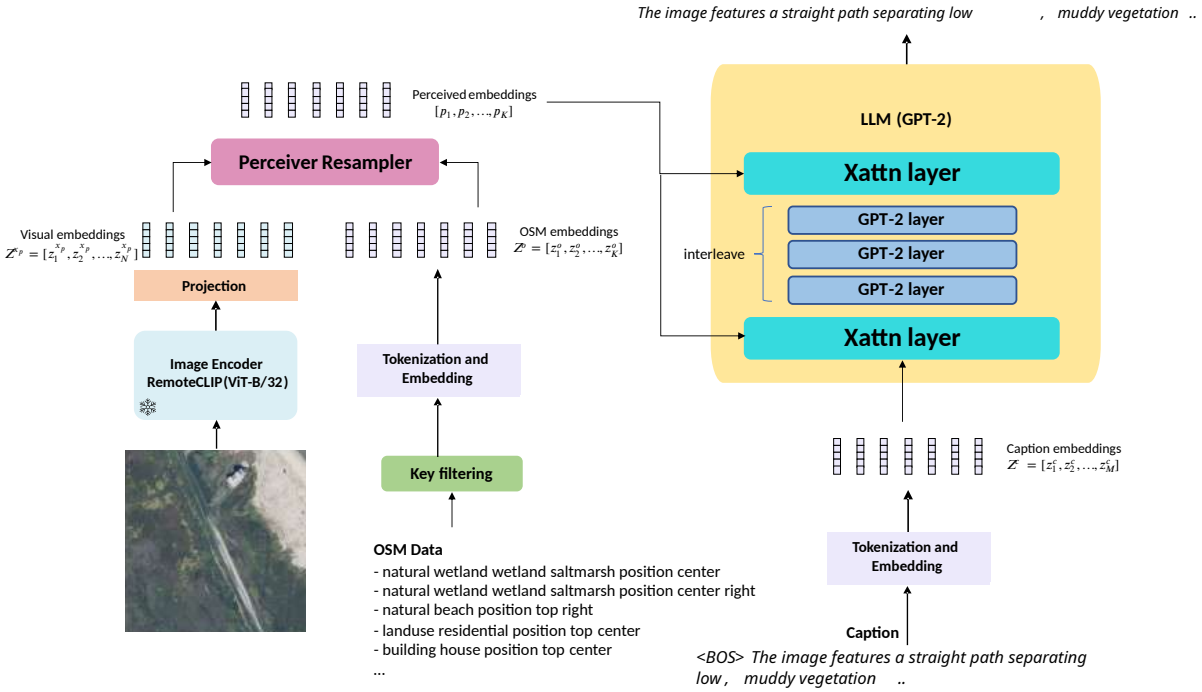


Figure 7.5: Overview of our model. The image is processed by a frozen image encoder that converts it into a set of visual embeddings Z_x. The additional data from OSM is filtered, concatenated in a single text string, tokenized and embedded into Z_o. The two sequences are concatenated and processed by a Perceiver Resampler module, which extracts a fixed set of "perceived" embeddings Z_p, which are used in cross-attention (Xattn) layers inside the language model to inject visual and OSM information for caption generation.

modules. Then, the visual embedding of the image and the OSM embeddings are forwarded to a Perceiver Resampler module, as described in [170]. This module contains a *fixed* number of learnable embeddings Z^p that interact with the input embeddings through a series of cross-attention layers to condense the information in the input sequence into a smaller and fixed set of embeddings. The resulting "perceived" embeddings Z^{x_p} are used inside a frozen language model (GPT-2), where cross-attention (Xattn) layers are interleaved with standard fixed layers. Cross-attention layers are initialized with the fixed language model's layers, but their output representation is updated through cross-attention between the caption embeddings from the previous layer Z^c_{l-1} and the perceived embeddings Z^{x_p}. Furthermore, cross-attention layers contain a gating mechanism, learned during training, to control the flow of information from the perceived embeddings. This way, if the gate is manually set to zero, cross-attention (Xattn) layers become standard GPT-2 layers, leaving the pre-trained GPT-2 network untouched.

Preliminary results have been focused on analyzing the effect of the inclusion of OSM data on caption generation. We tested two different configurations, using either the general or the augmented version of the captions. We trained a model for each configuration using only the image as input and another model using both the image and the associated OSM data. When using only the image, an attention mask is used inside the Perceiver Resampler to mask out OSM embeddings. We fixed the random seed to ensure that any changes are solely due to the different data sources, avoiding variations caused by different weight initializations or dataset shuffling during training. We measure the performance using Bleu-4 and Perplexity. Perplexity is a measure often used to evaluate the language modeling ability of an

LLM. It measures the uncertainty of the model in predicting the next token in a sequence. It is usually reported in aggregated form, calculated over several sequences (such as those in the validation set). For a single sequence of length N, perplexity is calculated as

$$\text{Perplexity} = \exp\Big(-\sum_{i=1}^{N} \log p(t_i \mid t_1, t_2, \ldots, t_{i-1}) \Big) \tag{7.1}$$

where t_i are single tokens in the sequence. As can be seen, the term inside the summation is the cross entropy loss between the probability of the true token $p(t_i) = 1$ and the probability assigned by the language model $p(t_i \mid t_1, t_2, \ldots, t_{i-1})$. Thus, when training an LLM using cross-entropy loss, perplexity can be calculated as the exponential of the loss.

7.4.2 Experimental setup

Our model uses the RemoteCLIP-ViT-B/32 [94] pre-trained image encoder and the GPT-2 base pre-trained LLM. We use a batch size of 8 and a weight decay of $1e^{-3}$. The Perceiver Resampler has 4 layers, with 16 learnable embeddings Z^p of dimension $d_p = 768$. The interleave between Xattn and GPT-2 layers is 1, meaning layers inside GPT-2 alternate between standard GPT-2 and Xattn layers. We trained our models for 15 epochs, monitoring the validation perplexity. We keep the best model as the one reaching the lowest validation perplexity during training.

7.5 Results and discussion

Figure 7.6 shows the progress of the model's perplexity during training. The perplexity of the training and validation sets decreased over the first three epochs. However, signs of over-fitting emerge by the fourth epoch: while the training perplexity decreases, the validation perplexity rises. The model exhibits similar behavior when using either the image alone or combined with OSM data as input. The training curves are nearly identical at the beginning of training and slightly diverge toward the end. This suggests that in our current model, including OSM data is ineffective, as it does not affect the training speed or the convergence. Tables 7.5 and 7.6 reinforce this observation. The Bleu-4 scores for the model trained on images alone and the model trained with both data sources are very similar. This indicates that the current method of integrating OSM data is ineffective, as the model fails to capture meaningful dependencies between the two modalities to produce better captions. The qualitative results, shown in

		Perplexity		**Bleu-4**	
		Train	**Test**	**Train**	**Test**
Best Model	img	6.75	7.99	0.14	0.13
	img+osm	5.95	8.00	0.14	0.13
Overfitted Model	img	2.66	10.90	0.24	0.13
	img+osm	2.58	11.39	0.27	0.13

Table 7.5: Quantitative results on **general** caption prediction. Best model: lowest perplexity on val. Overfitted model: last epoch checkpoint. img: only the image is used as input. img+osm: both image and OSM data as input.

Figures 7.8 and 7.9, reveal that our model struggles to incorporate the additional information provided by OSM tags. Indeed, despite the tags indicating the presence of a golf course, the model fails to recognize it. This limitation is also visible when the model receives only the image in input, suggesting that the

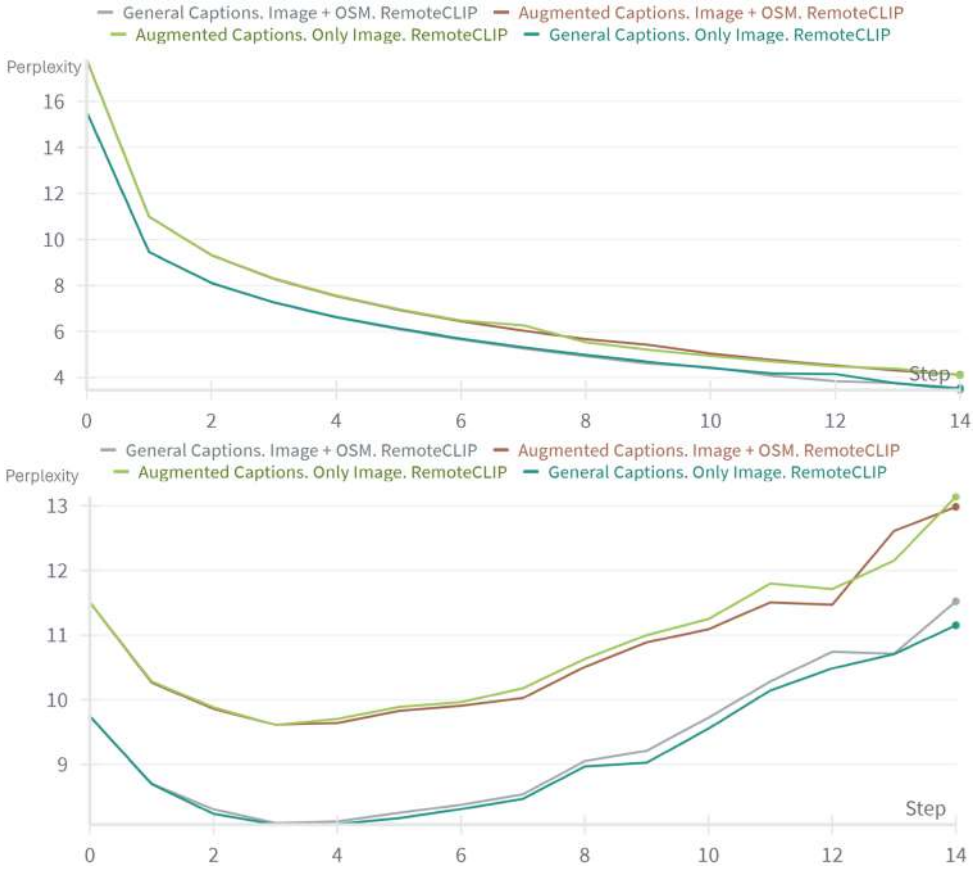


Figure 7.6: Training (top) and validation (bottom) perplexity of the proposed model when predicting for different inputs (only image or image+OSM) and different targets (general and augmented captions).

		Perplexity		**Bleu-4**	
		Train	**Test**	**Train**	**Test**
Best Model	img	6.75	10.20	0.14	0.13
	img+osm	6.73	10.21	0.15	0.13
Overfitted Model	img	3.11	14.29	0.22	0.15
	img+osm	3.13	14.44	0.23	0.15

Table 7.6: Quantitative results on **augmented** caption prediction. Best model: lowest perplexity on val. Overfitted model: last epoch checkpoint. img: only the image is used as input. img+osm: both image and OSM data as input.

challenges may stem from our long and detailed captions, which represent a difficult target. We think that the modest size of our dataset and the fact that the Perceiver Resampler is trained from scratch are exacerbating this difficulty.

Analyzing the unsatisfactory performance of our model, we identified some critical aspects that need

to be further analyzed:

1. **Different number of OSM tags for different images**: as shown in Figure 7.7, the distribution of OSM tags associated with each image exhibits significant variation rather than being centered around a specific value. This distribution shift due to the differing number of tags results in longer or shorter sequences of OSM embeddings that the Perceiver Resampler must learn to handle. This is challenging, as the Perceiver Resampler needs to learn how to compress sequences of different lengths into the same set of fixed learnable tokens. This can also explain why the model shows so little difference between using only the image or both data sources. We hypothesize that since the sequence of visual tokens coming from the image is always of fixed length, it represents a more stable input for the Perceiver Resampler, which thus tends to extract information more coherently from the visual data source. To address this limitation, we need a mechanism to filter the most relevant OSM data for a given image. We think that reducing the variability in the number of OSM tags would enhance its integration with the visual information.

2. **Perceiver Resampler trained from scratch**: Perceiver Resampler is trained from scratch in our architecture. However, to address the challenge of limited data, we believe it would be beneficial to leverage pre-trained components as much as possible. One possibility would be integrating the OSM data directly into the language model input, thus bypassing additional computations. However, this poses a risk of generating excessively long sequences that could exceed the language model's maximum sequence length, leading to issues during both training and inference. This approach would become more feasible after developing a reliable method to filter the most relevant OSM tags.

3. **Limited dataset**: our dataset is limited in size due to the difficulty and time-consuming nature of labeling additional samples. One potential approach to augment the labeled data could be leveraging semi-supervised methods, such as fine-tuning existing vision-language models on our dataset to generate additional samples. These synthetic captions could then be manually reviewed for accuracy. However, this process risks contaminating our dataset, which is currently curated with meticulous human annotation.

4. **Using pre-trained vision-language models**: an alternative approach is to take advantage of the capabilities of large pre-trained vision-language models. As discussed in Chapter 2, these models are highly sample-efficient and can learn complex tasks with significantly less labeled data than models trained from scratch. However, their large parameter sizes can pose challenges for training, particularly in resource-constrained environments.

7.6 Conclusion and future endeavors

This chapter explored the integration of ancillary geographic information into a remote-sensing image captioning pipeline. Specifically, we incorporated additional data from OpenStreetMap (OSM) into the model, enabling it to use both the image and the OSM data as complementary data streams during caption generation. Additionally, the geographic information allows the model to access specific details of the objects inside a scene, which cannot be inferred from the image alone, such as place names or building purposes. We manually labeled a dataset comprising image-caption-OSM tag triplets to investigate this possibility. Two captions were created: a *general* version, containing information deducible solely from visual content, and an *augmented* version, incorporating inside the general caption details obtainable only through OSM data. We build a model following the encoder-decoder architecture, inserting an additional module for OSM data integration. The OSM tags, treated as text, are concatenated into a single string

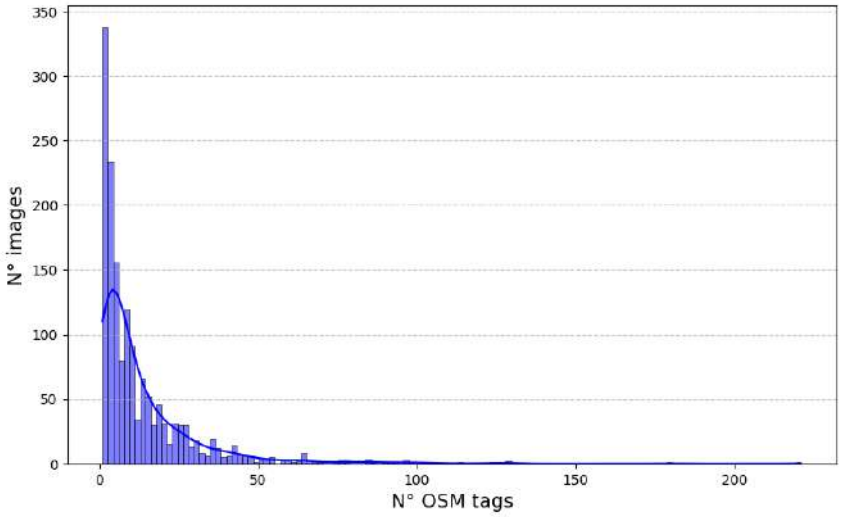


Figure 7.7: Distribution of the number of OSM tags per image in our dataset.

and processed through a Perceiver Resampler, which condenses the information from the image and OSM tags into a fixed set of learnable embeddings. The output of the Perceiver Resampler feeds into a GPT-2 language model decoder to generate the caption. We trained two models, one on general and the other on augmented descriptions. The results indicate that the model is severely overfitting on the training distribution, thus failing to effectively leverage the complementary information from the image and the OSM data. The captioning performance is unsatisfactory, and further research is needed to build a better model. We identified potential reasons for the limited performance and proposed future research directions to build a more robust model capable of effectively integrating geographic and visual data. We believe that incorporating ancillary geographic information is a promising avenue for advancing remote-sensing vision-language models, particularly in developing systems that are more grounded in the specific details of a scene.

General Caption:
The aerial view shows a golf course with forested areas, featuring a small dark pond positioned in the center right. Paths are scattered around the golf course.
Augmented Caption:
Large areas of trees are visible covering distinct parts of the image. The vegetation is dense and the foliage is green or dark green. In between these patches of forest, there is a golf course called "Azalea City Golf Course". In the center right part, a little body of water is visible, with dark waters. Small paths are visible, probably used to access the golf course."

OSM tags
leisure: golf_course, name: Azalea City Golf Course, position: center right
golf: water_hazard, natural: water, position: center right
leisure: park, name: Langan Park, position: bottom right
golf: green, landuse: grass, leisure: pitch, sport: golf, position: center right
golf: bunker, natural: sand, position: center left
golf: tee, landuse: grass, position: top left

Predictions:

img (gen)
A thick forest on the left side, while a river flows diagonally from the center left to the top right. In the bottom right corner, there is a building with a white roof.

img+osm (gen)
A landscape with a winding river flowing through the center. On the left side, there is a thick forest, while on the right side, a small pond is visible.

img (aug)
The image shows a large expanse of green grassland. In the top left corner, there is a small section of a dry area.

img+osm (aug)
The image shows a large area of land with a few scattered green plants. In the top left corner, there is a small area of bare soil.

Figure 7.8: Qualitative result of our best model on a random test image. (gen): model trained on general captions. (aug): model trained on augmented captions. Some OSM tags have been omitted for clarity.

General Caption:
The aerial view shows a golf course with forested areas, featuring a small dark pond positioned in the center right. Paths are scattered around the golf course.
Augmented Caption:
Large areas of trees are visible covering distinct parts of the image. The vegetation is dense and the foliage is green or dark green. In between these patches of forest, there is a golf course called "Azalea City Golf Course". In the center right part, a little body of water is visible, with dark waters. Small paths are visible, probably used to access the golf course."

OSM tags
leisure: golf_course, name: Azalea City Golf Course, position: center right
golf: water_hazard, natural: water, position: center right
leisure: park, name: Langan Park, position: bottom right
golf: green, landuse: grass, leisure: pitch, sport: golf, position: center right
golf: bunker, natural: sand, position: center left
golf: tee, landuse: grass, position: top left

Predictions:

img (gen)
A landscape with sparse vegetation and some bare patches. At the top, there is a horizontal road. In the bottom left corner, a small white building can be seen.

img+osm (gen)
A landscape with sparse vegetation and bare soil. A road runs horizontally through the center of the scene.

img (aug)
The image shows a dry landscape with sparse vegetation scattered throughout. In the center, there is a horizontal road that runs across the scene.

img+osm (aug)
The image shows a dry landscape featuring sparse vegetation and dry grass. In the top left corner, there is a light path that runs horizontally through the scene.

Figure 7.9: Qualitative result of the over-fitted model on a random test image. (gen): model trained on general captions. (aug): model trained on augmented captions. Some OSM tags have been omitted for clarity.

Conclusions and future directions

This thesis explored several strategies to improve critical aspects regarding remote sensing image captioning. After a brief introduction to some key concepts about generative image captioning in Chapter 2, Chapter 3 focuses on methods to enhance robustness and accuracy. The first proposed approach introduced an ensemble-based image captioning framework that integrates outputs from diverse expert models. By employing various fusion strategies, this framework combines captions into a single, more robust output. To improve accuracy, we tested fine-tuning a pre-trained large vision-language model with a novel multi-task instruction dataset. This dataset combined two image captioning and two visual question-answering datasets specifically tailored for remote sensing. The study revealed that integrating different tasks and datasets is advantageous only when the dataset's images share common characteristics. In contrast, when the images differ significantly—such as being captured at varying resolutions or angles—the shared information between datasets and tasks can hinder performance. Furthermore, the analysis revealed the high sample efficiency of pre-trained large vision-language models, which on single tasks, already obtain strong results, sometimes resulting in state-of-the-art performance.

Chapter 4 explored strategies to enhance the detail and expressiveness of captions. A notable challenge identified was that many existing remote-sensing image captioning datasets often assign identical, concise captions to visually distinct scenes. This can not only hinder text-to-image retrieval systems from discriminating effectively between scenes but also fails to provide end users with sufficiently detailed information. To overcome this limitation without the need for additional, more detailed captions, which are costly and time-consuming to obtain, we propose and test a novel approach leveraging a simulated visual dialogue. This method involves an interactive exchange between two machines: a questioner, designed to generate queries to explore the visual content, and an answerer, which answers the questions based on the visual content. We show that our machine-to-machine visual dialogue (M2MVD) approach significantly enhances caption richness and improves the ability to differentiate between visually similar scenes.

Chapter 5 analyzes some limitations in the question-generation process of our visual dialogue system by delving into the task of Visual Question Generation (VQG) for remote sensing imagery. VQG aims to create coherent and meaningful questions tailored to a specific remote sensing scene. We noticed that existing datasets in RS-VQG literature are largely answer-centric, leading to template-based questions that often lack diversity and naturalness. Thus we first introduced a novel dataset comprising human-labeled questions that are richer and more varied, and then trained a vision-language model to generate an entire paragraph of questions for a given image. Our analysis of the model's outputs revealed that it effectively generates questions grounded in the visual content.

Chapter 6 explores the temporal dimension by extending our dialogue-based approach to describe changes between pairs of images. To account for multiple images, we leverage a pre-trained vision-language model capable of processing and solving tasks involving sequences of images. We reveal how directly solving this task requires a level of specialization not yet available in open-source pre-trained vision-language models. However, we explore directing the question-generation model to focus on specific areas of interest. This approach enables the creation of more precise and relevant descriptions tailored to the user's needs. To evaluate the quality of the outputs, we introduce a novel metric called FMScore,

which assesses whether a series of facts is corroborated by a given piece of text. This metric serves as an alternative to traditional word-matching reference metrics in NLP, which are unsuitable for evaluating outputs from pre-trained vision-language models. We show that although pre-trained vision-language models struggle with directly solving this task, our dialogue-based paradigm shows potential by simplifying the problem into a sequence of more manageable sub-tasks.

Chapter 7 introduces our initial efforts to integrate additional geographic information into remote-sensing vision-language tasks. We focus on image captioning and present a new dataset with descriptions enriched by specific details derived from OpenStreetMap (OSM) tags related to geographic features in the scene. We think that adding specific scene-related details helps make vision-language models more grounded and reliable, especially for remote sensing tasks. However, combining multiple data sources within a single framework poses unique challenges. These challenges arise from the high variability in OSM data within individual samples and the limited number of available samples due to the cost involved in manual labeling. We identify some key challenges and propose future research directions to improve the integration of diverse data sources and develop models that effectively use their complementary strengths.

Then, after three years, I'm here looking back at the transformation that happened to my research field. At the start of my PhD, remote sensing vision-language tasks were mainly tackled using specialized models trained on single tasks. The potential of large language models as multi-task agents was not evident, and conversational agents were not yet available. During the last three years, the paradigm has shifted from models trained on single tasks to models that can understand and solve multiple tasks at once. This shift has also influenced my perspective on the field. In the following, I will discuss my thoughts on the future of this area, as well as the broader evolution of remote sensing vision-language tasks.

Nowadays, a series of pre-trained models are available open-source: both language only [112][171][172] and vision-language [104][164][170]. The majority of those models are specialized in natural scenes. Yet, the pre-training step using large amounts of data makes them highly sample efficient during transfer learning, even when applied to a downstream task that involves a drastically different distribution (such as when fine-tuning them on remote sensing datasets). We saw in Chapter 3 how fine-tuning a pre-trained large vision language model (LLaVA) on single tasks (image captioning or VQA) produces strong baselines, surpassing previous state-of-the-art specialized architectures for the field. On the same line, I believe that future efforts in remote sensing vision-language tasks should shift from single highly specialized models to general models that can solve several tasks at once, using language as an interface. Some works in this direction are already starting to emerge in the remote sensing field [60][58]. Toward this goal, there is a strong need to have remote sensing foundational models specifically trained on remote sensing data in unsupervised ways. Efforts in this direction [173] could accelerate performance gains by removing the distribution gap between general web-scraped visual-text data (used today to pre-train those large models) and remote sensing-related data.

Another aspect to take into consideration is the use of instruction-following models. As we saw in Chapter 2, these are models that have been specifically fine-tuned from pre-trained models to generate different outputs based on user instructions. As demonstrated in Chapter 6, these models can be guided through textual prompts to emphasize specific elements of interest. I believe that this capability is particularly valuable in remote-sensing applications, where the primary focus often lies on specific aspects of the data rather than exhaustive analysis of all available information. For instance, consider a scenario in which a user seeks to identify and localize swimming pools within an aerial image. If the user's interest subsequently shifts to localizing baseball fields, having a model that can adapt and change the output based on the request can eliminate the need to develop and deploy distinct models for each task, enhancing efficiency.

Moreover, advancements in *computational* efficiency are crucial, particularly in the context of large language models (LLMs) and large vision-language models, which contain billions of parameters. These models are often impractical for real-time applications or deployment on resource-constrained devices.

Thus, exploring strategies to reduce model complexity while maintaining performance is vital for facilitating their broader use in real-world remote sensing applications. Several initiatives are addressing this challenge [174] [175] [172], introducing compact language models with performance comparable to that of much larger models.

Lastly, we believe that remote-sensing applications should evolve from focusing solely on images to integrating additional data sources. GIS databases, which provide grounded and specific information, represent a valuable resource that must not be overlooked in future developments. We propose that a vision-language assistant designed to address remote-sensing tasks would greatly benefit from incorporating ancillary geographic information, thereby improving both the specificity and the groundedness of its responses.

Bibliography

[1] Aditya Vailaya, Anil Jain, and Hong Jiang Zhang. "On image classification: City images vs. landscapes". In: *Pattern recognition* 31.12 (1998), pp. 1921–1935.

[2] Martin Szummer and Rosalind W Picard. "Indoor-outdoor image classification". In: *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*. IEEE. 1998, pp. 42–51.

[3] Anna Bosch, Andrew Zisserman, and Xavier Munoz. "Scene classification via pLSA". In: *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9*. Springer. 2006, pp. 517–530.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[5] Francisco Herrera et al. *Multilabel classification*. Springer, 2016.

[6] Matthew R Boutell et al. "Learning multi-label scene classification". In: *Pattern recognition* 37.9 (2004), pp. 1757–1771.

[7] P Viola and M Jones. "Rapid object detection using a boosted cascade of simple features, CVPR 2001. PJ Phillips, M. Hyeonjoon, et. al., The FERET Evaluation Methodology for Face-Recognition Algorithms". In: *IEEE Trans. on. PAMI* 22.10 (2000), pp. 1090–1104.

[8] Jianguo Wang and Tieniu Tan. "A new face detection method based on shape information". In: *Pattern Recognition Letters* 21.6-7 (2000), pp. 463–471.

[9] Zhengxia Zou et al. "Object Detection in 20 Years: A Survey". In: *Proceedings of the IEEE* 111.3 (2023), pp. 257–276. DOI: 10.1109/JPROC.2023.3238524.

[10] Shervin Minaee et al. "Image segmentation using deep learning: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3523–3542.

[11] Shunli Wang et al. "Category attention guided network for semantic segmentation of Fine-Resolution remote sensing images". In: *International Journal of Applied Earth Observation and Geoinformation* 127 (2024), p. 103661.

[12] Genc Hoxha, Farid Melgani, and Begüm Demir. "Retrieving Images with Generated Textual Descriptions". In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. 2019, pp. 5812–5815. DOI: 10.1109/IGARSS.2019.8899321.

[13] Matan Levy et al. "Chatting Makes Perfect: Chat-based Image Retrieval". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 61437–61449.

[14] Qingtian Zeng, Jian Sun, and Shansong Wang. "DIC-Transformer: interpretation of plant disease classification results using image caption generation technology". In: *Frontiers in Plant Science* 14 (2024), p. 1273029.

[15] Wei Sun et al. “Veg-DenseCap: Dense captioning model for vegetable leaf disease images”. In: *Agronomy* 13.7 (2023), p. 1700.

[16] Övgü Özdemir and Erdem Akagündüz. “Enhancing Visual Question Answering through Question-Driven Image Captions as Prompts”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 1562–1571.

[17] Soravit Changpinyo et al. “All you may need for vqa are image captions”. In: *arXiv preprint arXiv:2205.01883* (2022).

[18] Wei Li et al. “The traffic scene understanding and prediction based on image captioning”. In: *IEEE Access* 9 (2020), pp. 1420–1427.

[19] Hareem Ayesha et al. “Automatic medical image interpretation: State of the art and future directions”. In: *Pattern Recognition* 114 (2021), p. 107856.

[20] John Pavlopoulos, Vasiliki Kougia, and Ion Androutsopoulos. “A survey on biomedical image captioning”. In: *Proceedings of the second workshop on shortcomings in vision and language*. 2019, pp. 26–36.

[21] Hiba Ahsan et al. “Multi-modal image captioning for the visually impaired”. In: *arXiv preprint arXiv:2105.08106* (2021).

[22] Danna Gurari et al. “Captioning images taken by people who are blind”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer. 2020, pp. 417–434.

[23] Pierre Dognin et al. “Image captioning as an assistive technology: Lessons learned from vizwiz 2020 challenge”. In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 437–459.

[24] Thao Nguyen et al. “Improving multimodal datasets with image captioning”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[25] Rémi Lebret, Pedro Pinheiro, and Ronan Collobert. “Phrase-based image captioning”. In: *International conference on machine learning*. PMLR. 2015, pp. 2085–2094.

[26] Siming Li et al. “Composing simple image descriptions using web-scale n-grams”. In: *Proceedings of the fifteenth conference on computational natural language learning*. 2011, pp. 220–228.

[27] Ali Farhadi et al. “Every picture tells a story: Generating sentences from images”. In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer. 2010, pp. 15–29.

[28] Micah Hodosh, Peter Young, and Julia Hockenmaier. “Framing image description as a ranking task: Data, models and evaluation metrics”. In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 853–899.

[29] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. “Im2text: Describing images using 1 million captioned photographs”. In: *Advances in neural information processing systems* 24 (2011).

[30] Richard Socher et al. “Grounded compositional semantics for finding and describing images with sentences”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 207–218.

[31] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. “Deep fragment embeddings for bidirectional image sentence mapping”. In: *Advances in neural information processing systems* 27 (2014).

[32] Junhua Mao et al. “Explain images with multimodal recurrent neural networks”. In: *arXiv preprint arXiv:1410.1090* (2014).

[33] Oriol Vinyals et al. “Show and tell: A neural image caption generator”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.

[34] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[35] Bo Qu et al. "Deep semantic understanding of high resolution remote sensing image". In: *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*. 2016, pp. 1–5. DOI: 10.1109/CITS.2016.7546397.

[36] Xiaoqiang Lu et al. "Exploring Models and Data for Remote Sensing Image Caption Generation". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.4 (2018), pp. 2183–2195. DOI: 10.1109/TGRS.2017.2776321.

[37] Xiangqing Shen et al. "Remote sensing image captioning via variational autoencoder and reinforcement learning". In: *Knowledge-Based Systems* 203 (2020), p. 105920.

[38] Zhengxin Li et al. "Cross-modal retrieval and semantic refinement for remote sensing image captioning". In: *Remote Sensing* 16.1 (2024), p. 196.

[39] Xueting Zhang et al. "Multi-scale cropping mechanism for remote sensing image captioning". In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 10039–10042.

[40] Yangyang Li et al. "A multi-level attention model for remote sensing image captions". In: *Remote Sensing* 12.6 (2020), p. 939.

[41] Qimin Cheng et al. "NWPU-captions dataset and MLCA-net for remote sensing image captioning". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–19.

[42] Chengze Wang, Zhiyu Jiang, and Yuan Yuan. "Instance-aware remote sensing image captioning with cross-hierarchy attention". In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2020, pp. 980–983.

[43] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA, 2017, pp. 6000–6010. ISBN: 9781510860964.

[44] Junjue Wang et al. "Capformer: pure transformer for remote sensing image caption". In: *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2022, pp. 7996–7999.

[45] Usman Zia, M Mohsin Riaz, and Abdul Ghafoor. "Transforming remote sensing images to textual descriptions". In: *International Journal of Applied Earth Observation and Geoinformation* 108 (2022), p. 102741.

[46] Runyan Du et al. "From plane to hierarchy: Deformable transformer for remote sensing image captioning". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2023).

[47] Yinan Wu et al. "TrTr-CMR: Cross-Modal Reasoning Dual Transformer for Remote Sensing Image Captioning". In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[48] Kai Zhao and Wei Xiong. "Exploring region features in remote sensing image captioning". In: *International Journal of Applied Earth Observation and Geoinformation* 127 (2024), p. 103672.

[49] Zhengyuan Zhang et al. "LAM: Remote sensing image captioning with label-attention mechanism". In: *Remote Sensing* 11.20 (2019), p. 2349.

[50] Kangda Cheng et al. "Remote Sensing Image Captioning with Multi-Scale Feature and Small Target Attention". In: *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2024, pp. 7436–7439.

[51] Hitesh Kandala et al. “Exploring transformer and multilabel classification for remote sensing image captioning”. In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5.

[52] Xiutiao Ye et al. “A joint-training two-stage method for remote sensing image captioning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–16.

[53] Qiaoqiao Yang, Zihao Ni, and Peng Ren. “Meta captioning: A meta learning based remote sensing image captioning framework”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 186 (2022), pp. 190–200.

[54] Nirmala Murali and AP Shanthi. “Remote sensing image captioning via multilevel attention-based visual question answering”. In: *Innovations in Computational Intelligence and Computer Vision: Proceedings of ICICV 2021*. Springer, 2022, pp. 465–475.

[55] Yingxu He and Qiqi Sun. “Towards Automatic Satellite Images Captions Generation Using Large Language Models”. In: *arXiv preprint arXiv:2310.11392* (2023).

[56] João Daniel Silva et al. “Large language models for captioning and retrieving remote sensing images”. In: *arXiv preprint arXiv:2402.06475* (2024).

[57] Yang Zhan, Zhitong Xiong, and Yuan Yuan. “Skyeyegpt: Unifying remote sensing vision-language tasks via instruction tuning with large language model”. In: *arXiv preprint arXiv:2401.09712* (2024).

[58] Kartik Kuckreja et al. “Geochat: Grounded large vision-language model for remote sensing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 27831–27840.

[59] Yuan Hu et al. “Rsgpt: A remote sensing vision language model and benchmark”. In: *arXiv preprint arXiv:2307.15266* (2023).

[60] Wei Zhang et al. “Earthgpt: A universal multi-modal large language model for multi-sensor image comprehension in remote sensing domain”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[61] Philip Gage. “A new algorithm for data compression”. In: *The C Users Journal* 12.2 (1994), pp. 23–38.

[62] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[63] Ari Holtzman et al. “The curious case of neural text degeneration”. In: *arXiv preprint arXiv:1904.09751* (2019).

[64] Ibomoiye Domor Mienye, Theo G Swart, and George Obaido. “Recurrent neural networks: A comprehensive review of architectures, variants, and applications”. In: *Information* 15.9 (2024), p. 517.

[65] S Hochreiter. “Long Short-term Memory”. In: *Neural Computation MIT-Press* (1997).

[66] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[67] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.0473.

[68] Kelvin Xu et al. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 2048–2057.

[69] Jiasen Lu et al. “Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[70] Bryan McCann et al. “The natural language decathlon: Multitask learning as question answering”. In: *arXiv preprint arXiv:1806.08730* (2018).

[71] Rishi Bommasani et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).

[72] Tom Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[73] Jared Kaplan et al. “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361* (2020).

[74] Long Ouyang et al. *Training language models to follow instructions with human feedback*. 2022. arXiv: 2203.02155 [cs.CL].

[75] Kishore Papineni et al. “Bleu: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[76] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. “Cider: Consensus-based image description evaluation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4566–4575.

[77] Satanjeev Banerjee and Alon Lavie. “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”. In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[78] Lin Chin-Yew. “Rouge: A package for automatic evaluation of summaries”. In: *Proceedings of the Workshop on Text Summarization Branches Out, 2004*. 2004.

[79] Xinlei Chen et al. “Microsoft COCO Captions: Data Collection and Evaluation Server”. In: *CoRR* abs/1504.00325 (2015). eprint: 1504.00325.

[80] Peter Young et al. “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 67–78. DOI: 10.1162/tacl_a_00166.

[81] J. Kittler et al. “On combining classifiers”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.3 (1998), pp. 226–239. DOI: 10.1109/34.667881.

[82] G.J. Briem, J.A. Benediktsson, and J.R. Sveinsson. “Multiple classifiers applied to multisource remote sensing data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 40.10 (2002), pp. 2291–2299. DOI: 10.1109/TGRS.2002.802476.

[83] F. Melgani and Y. Bazi. “Markovian Fusion Approach to Robust Unsupervised Change Detection in Remotely Sensed Imagery”. In: *IEEE Geoscience and Remote Sensing Letters* 3.4 (2006), pp. 457–461. DOI: 10.1109/LGRS.2006.875773.

[84] Harshitha Katpally and Ajay Bansal. “Ensemble Learning on Deep Neural Networks for Image Caption Generation”. In: *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*. 2020, pp. 61–68. DOI: 10.1109/ICSC.2020.00016.

[85] Chenyang Liu, Rui Zhao, and Zhenwei Shi. "Remote-Sensing Image Captioning Based on Multilayer Aggregated Transformer". In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5. DOI: 10.1109/LGRS.2022.3150957.

[86] Junnan Li et al. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 19730–19742.

[87] David Nukrai, Ron Mokady, and Amir Globerson. "Text-Only Training for Image Captioning using Noise-Injected CLIP". In: *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. 2022, pp. 4055–4063. DOI: 10.18653/V1/2022.FINDINGS-EMNLP.299.

[88] Hyung Won Chung et al. "Scaling Instruction-Finetuned Language Models". In: *Journal of Machine Learning Research* 25.70 (2024), pp. 1–53.

[89] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[90] Wenhui Wang et al. "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5776–5788.

[91] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[92] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2021.

[93] Huggingface. *flax-community/clip-rsicd-v2*. URL: https://huggingface.co/flax-community/clip-rsicd-v2.

[94] Fan Liu et al. "RemoteCLIP: A Vision Language Foundation Model for Remote Sensing". In: *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), pp. 1–16. DOI: 10.1109/TGRS.2024.3390838.

[95] Zilun Zhang et al. "RS5M and GeoRSCLIP: A large scale vision-language dataset and a large vision-language model for remote sensing". In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[96] Samuel R. Bowman et al. "Generating Sentences from a Continuous Space". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. 2016, pp. 10–21. DOI: 10.18653/v1/K16-1002.

[97] Chunyuan Li et al. "Optimus: Organizing sentences via pre-trained modeling of a latent space". In: *arXiv preprint arXiv:2004.04092* (2020).

[98] Michael Crawshaw. "Multi-task learning with deep neural networks: A survey". In: *arXiv preprint arXiv:2009.09796* (2020).

[99] Jun Yu et al. "Unleashing the Power of Multi-Task Learning: A Comprehensive Survey Spanning Traditional, Deep, and Pretrained Foundation Model Eras". In: *arXiv preprint arXiv:2404.18961* (2024).

[100] Zhanpeng Zhang et al. "Facial landmark detection by deep multi-task learning". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer. 2014, pp. 94–108.

[101] Xiaodong Liu et al. “Representation learning using multi-task deep neural networks for semantic classification and information retrieval”. In: (2015).

[102] Ting Chen et al. “Pix2seq: A language modeling framework for object detection”. In: *arXiv preprint arXiv:2109.10852* (2021).

[103] Zhaowei Cai et al. “X-detr: A versatile architecture for instance-wise vision-language tasks”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 290–308.

[104] Haotian Liu et al. “Visual instruction tuning”. In: *Advances in neural information processing systems* 36 (2024).

[105] Hugo Touvron et al. “Llama: Open and efficient foundation language models”. In: *arXiv preprint arXiv:2302.13971* (2023).

[106] Junnan Li et al. “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation”. In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 12888–12900.

[107] Edward J Hu et al. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).

[108] Genc Hoxha and Farid Melgani. “A novel SVM-based decoder for remote sensing image captioning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–14.

[109] Sylvain Lobry et al. “RSVQA: Visual question answering for remote sensing data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.12 (2020), pp. 8555–8566.

[110] Xiangtao Zheng et al. “Mutual attention inception network for remote sensing visual question answering”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–14.

[111] Gui-Song Xia et al. “DOTA: A large-scale dataset for object detection in aerial images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3974–3983.

[112] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. 2023. URL: `https://lmsys.org/blog/2023-03-30-vicuna/`.

[113] Binqiang Wang et al. “Semantic descriptions of high-resolution remote sensing images”. In: *IEEE Geoscience and Remote Sensing Letters* 16.8 (2019), pp. 1274–1278.

[114] Zhengyuan Zhang et al. “VAA: Visual aligning attention model for remote sensing image captioning”. In: *IEEE Access* 7 (2019), pp. 137355–137364.

[115] Zhenghang Yuan, Xuelong Li, and Qi Wang. “Exploring multi-level attention and semantic relationship for remote sensing image captioning”. In: *IEEE Access* 8 (2019), pp. 2608–2620.

[116] Wei Huang, Qi Wang, and Xuelong Li. “Denoising-based multiscale feature fusion for remote sensing image captioning”. In: *IEEE Geoscience and Remote Sensing Letters* 18.3 (2020), pp. 436–440.

[117] Gencer Sumbul, Sonali Nayak, and Begüm Demir. “SD-RSIC: Summarization-driven deep remote sensing image captioning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.8 (2020), pp. 6922–6934.

[118] Xuelong Li et al. “Truncation cross entropy loss for remote sensing image captioning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.6 (2020), pp. 5246–5257.

[119] Xiaofeng Ma, Rui Zhao, and Zhenwei Shi. “Multiscale methods for optical remote-sensing image captioning”. In: *IEEE Geoscience and Remote Sensing Letters* 18.11 (2020), pp. 2001–2005.

[120] Qi Wang et al. “Word–sentence framework for remote sensing image captioning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.12 (2020), pp. 10532–10543.

[121] Rui Zhao, Zhenwei Shi, and Zhengxia Zou. “High-resolution remote sensing image captioning based on structured attention”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–14.

[122] Yunpeng Li et al. “Recurrent attention and semantic gate for remote sensing image captioning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–16.

[123] Shuo Zhuang et al. “Improving remote sensing image captioning by combining grid features and transformer”. In: *IEEE Geoscience and Remote Sensing Letters* 19 (2021), pp. 1–5.

[124] Yong Wang et al. “Multiscale multiinteraction network for remote sensing image captioning”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 2154–2165.

[125] Sylvain Lobry et al. “RSVQA: Visual question answering for remote sensing data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.12 (2020), pp. 8555–8566.

[126] Zhenghang Yuan et al. “From easy to hard: Learning language-guided curriculum for visual question answering on remote sensing data”. In: *IEEE transactions on geoscience and remote sensing* 60 (2022), pp. 1–11.

[127] Yakoub Bazi et al. “Bi-modal transformer-based approach for visual question answering in remote sensing imagery”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–11.

[128] Yuan Hu et al. “Rsgpt: A remote sensing vision language model and benchmark”. In: *arXiv preprint arXiv:2307.15266* (2023).

[129] Abhishek Das et al. “Visual Dialog”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[130] Josh Achiam et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[131] Deyao Zhu et al. *ChatGPT Asks, BLIP-2 Answers: Automatic Questioning Towards Enriched Visual Descriptions*. 2023. arXiv: 2303.06594 [cs.CV].

[132] Yezhou Yang et al. “Neural self talk: Image understanding via continuous questioning and answering”. In: *arXiv preprint arXiv:1512.03460* (2015).

[133] Jack Hessel et al. “Clipscore: A reference-free evaluation metric for image captioning”. In: *arXiv preprint arXiv:2104.08718* (2021).

[134] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[135] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[136] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[137] Nasrin Mostafazadeh et al. “Generating natural questions about an image”. In: *arXiv preprint arXiv:1603.06059* (2016).

[138] Shijie Zhang et al. “Automatic generation of grounded visual questions”. In: *arXiv preprint arXiv:1612.06530* (2016).

[139] Taghreed Abdullah et al. “TextRS: Deep bidirectional triplet network for matching text to remote sensing images”. In: *Remote Sensing* 12.3 (2020), p. 405.

[140] Gui-Song Xia et al. "AID: A benchmark data set for performance evaluation of aerial scene classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7 (2017), pp. 3965–3981.

[141] Weixun Zhou et al. "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval". In: *ISPRS journal of photogrammetry and remote sensing* 145 (2018), pp. 197–209.

[142] Yi Yang and Shawn Newsam. "Bag-of-visual-words and spatial extensions for land-use classification". In: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. 2010, pp. 270–279.

[143] Gong Cheng, Junwei Han, and Xiaoqiang Lu. "Remote sensing image scene classification: Benchmark and state of the art". In: *Proceedings of the IEEE* 105.10 (2017), pp. 1865–1883.

[144] Stanislaw Antol et al. "Vqa: Visual question answering". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[145] Tong Wang, Xingdi Yuan, and Adam Trischler. "A joint model for question answering and question generation". In: *arXiv preprint arXiv:1706.01450* (2017).

[146] Unnat Jain, Ziyu Zhang, and Alexander G Schwing. "Creativity: Generating diverse questions using variational autoencoders". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6485–6494.

[147] Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. "Information maximizing visual question generation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2008–2018.

[148] Shagun Uppal et al. "C3VQG: Category consistent cyclic visual question generation". In: *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*. 2021, pp. 1–7.

[149] Harsh Jhamtani and Taylor Berg-Kirkpatrick. "Learning to Describe Differences Between Pairs of Similar Images". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 4024–4034. DOI: 10.18653/v1/D18-1436.

[150] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. "Robust change captioning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4624–4633.

[151] Xiangxi Shi et al. "Finding it at another side: A viewpoint-adapted matching encoder for change captioning". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, pp. 574–590.

[152] Hoeseong Kim et al. "Viewpoint-Agnostic change captioning with cycle consistency". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2095–2104.

[153] Yunbin Tu et al. "Semantic relation-aware difference representation learning for change captioning". In: *Findings of the association for computational linguistics: ACL-IJCNLP 2021*. 2021, pp. 63–73.

[154] Mehrdad Hosseinzadeh and Yang Wang. "Image change captioning by learning from an auxiliary task". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2725–2734.

[155] Yue Qiu et al. "Describing and localizing multiple changes with transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1971–1980.

[156] Shizhen Chang and Pedram Ghamisi. "Changes to captions: An attentive network for remote sensing change captioning". In: *IEEE Transactions on Image Processing* (2023).

[157] Genc Hoxha et al. "Change captioning: A new paradigm for multitemporal remote sensing image analysis". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–14.

[158] Chenyang Liu et al. "Remote sensing image change captioning with dual-branch transformers: A new method and a large scale dataset". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–20.

[159] Chen Cai, Yi Wang, and Kim-Hui Yap. "Interactive change-aware transformer network for remote sensing image change captioning". In: *Remote Sensing* 15.23 (2023), p. 5611.

[160] Chenyang Liu et al. "Progressive scale-aware network for remote sensing image change captioning". In: *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2023, pp. 6668–6671.

[161] Qing Zhou et al. "Single-stream Extractor Network with Contrastive Pre-training for Remote Sensing Change Captioning". In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[162] Chenyang Liu et al. "A decoupling paradigm with prompt learning for remote sensing image change captioning". In: *IEEE Transactions on Geoscience and Remote Sensing* (2023).

[163] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. 2023. URL: `https://lmsys.org/blog/2023-03-30-vicuna/`.

[164] Bo Li et al. "Otter: A Multi-Modal Model with In-Context Instruction Tuning". In: *arXiv preprint arXiv:2305.03726* (2023).

[165] Bo Li et al. "Mimic-it: Multi-modal in-context instruction tuning". In: *arXiv preprint arXiv:2306.05425* (2023).

[166] Kunping Yang et al. "Asymmetric siamese networks for semantic change detection in aerial images". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–18.

[167] Zhenghang Yuan et al. "Change detection meets visual question answering". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–13.

[168] Sofia Nikiforova et al. "Geo-aware image caption generation". In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 3143–3156.

[169] Xiang Li, Jian Ding, and Mohamed Elhoseiny. "VRSBench: A Versatile Vision-Language Benchmark Dataset for Remote Sensing Image Understanding". In: *arXiv preprint arXiv:2406.12384* (2024).

[170] Jean-Baptiste Alayrac et al. "Flamingo: a visual language model for few-shot learning". In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736.

[171] Abhimanyu Dubey et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[172] Marah Abdin et al. "Phi-3 technical report: A highly capable language model locally on your phone". In: *arXiv preprint arXiv:2404.14219* (2024).

[173] Johannes Jakubik et al. "Foundation models for generalist geospatial artificial intelligence". In: *CoRR* (2023).

[174] Peiyuan Zhang et al. "Tinyllama: An open-source small language model". In: *arXiv preprint arXiv:2401.02385* (2024).

[175] Jinze Bai et al. "Qwen technical report". In: *arXiv preprint arXiv:2309.16609* (2023).

List of Publications

[P1] Yakoub Bazi, Laila Bashmal, Mohamad Mahmoud Al Rahhal, Riccardo Ricci, and Farid Melgani. "Rs-llava: A large vision-language model for joint captioning and question answering in remote sensing imagery". In: *Remote Sensing* 16.9 (2024), p. 1477.

[P2] Riccardo Ricci, Farid Melgani, José Marcato Junior, and Wesley Nunes Gonçalves. "Robust Image Captioning with Post-Generation Ensemble Method". In: *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2023, pp. 5234–5237.

[P3] Riccardo Ricci, Farid Melgani, José Marcato Junior, and Wesley Nunes Gonçalves. "NLP-Based Fusion Approach to Robust Image Captioning". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2024).

[P4] Laila Bashmal, Yakoub Bazi, Farid Melgani, Riccardo Ricci, Mohamad M Al Rahhal, and Mansour Zuair. "Visual question generation from remote sensing images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 16 (2023), pp. 3279–3293.

[P5] Riccardo Ricci, Yakoub Bazi, and Farid Melgani. "Machine-to-machine visual dialoguing with ChatGPT for enriched textual image description". In: *Remote Sensing* 16.3 (2024), p. 441.

[P6] Riccardo Ricci, Yakoub Bazi, and Farid Melgani. "Change Captioning Meets Large Language and Vision Models". In: *Under revision for ISPRS Journal of Photogrammetry and Remote Sensing*. 2024.

[P7] Riccardo Ricci, Alberto Frizzera, Wesley Nunes Gonçalves, José Marcato Junior, and Farid Melgani. "Exploring Synthetic Captions for Remote Sensing Vision-Text Foundational Models". In: *2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS)*. IEEE. 2024, pp. 73–77.