Aerial image captioning with different visual backbones and GPT-2

Riccardo Lunelli, *Dipartimento di Ingengeria e Scienza dell' Informazione, Università degli studi di Trento*

***Abstract*—This work evaluates the performance of different visual encoders within an image captioning pipeline applied to aerial images. Specifically, I compaie the performance of VGG16 and CLIP as backbone encoders for extracting visual features, which are then integrated with a GPT-2 model to generate descriptive captions for aerial images.**

I. Introduction

The integration of computer vision and natural language processing has revolutionized the machine learning field. One important area of research in this domain is image captioning, where systems are trained to generate textual descriptions of images. This capability is very important for various applications, particularly in the field of remote sensing, where huge amounts of aerial and satellite imagery are collected daily. Unlike everyday photographs, aerial images capture a wide range of perspectives and often include specific details of urban environments, landscapes, and natural phenomena. Furthermore, these images require the ability to interpret complex spatial relationships and to identify objects and structures that may be unfamiliar to traditional image captioning models.

This project focuses on a pipeline consisting of three elements: an image encoder, an adaptation layer, and a decoder. The central point of this work is to analyze how different architectures and differently trained encoders influence the performance of a GPT-2 [6] decoder model. I employed three variants of encoders: the first is a VGG-16 [9] convolutional network, the second is a pre-finetuned version of CLIP [7], and the latter is the RemoteCLIP model [2].

II. Related works

The ability to combine features from different modalities has always been a critical problem in machine learning, posing a significant obstacle to multimodal tasks. One of the biggest advancements in this field was the development of the CLIP model by OpenAI. CLIP has two encoders: one for images and another for text descriptions. It utilizes a contrastive loss to maximize the similarity between the image and the corresponding caption while pushing apart non-matching image caption pairs. This approach ensures that the embedding from both encoders closely represent the same concept, despite the different modalities. CLIP was trained on a huge dataset of text-image pairs to learn a wide range of visual concepts with natural language supervision. This approach allows CLIP to perform zero-shot transfer learning, making it highly versatile and capable of generalizing to new tasks without requiring task-specific data. However, satellite images present a challenge due to their highly specialized and less frequently encountered visual data. As discussed by Radford et al., satellite images are not ideal candidates for zero-shot or few-shot learning settings. That is why for this domain it is usually fine-tuned on the specific dataset. In the field of remote sensing Liu et al. proposed a general-purpose vision-language foundation model for remote sensing, named RemoteCLIP. This model has the same architecture of CLIP but is trained on a huge collection of annotated data from 17 different datasets, with a total of almost 200.000 images and 5 caption per image. The author showed how this deeply trained model achieve state of the art abilities on zero-shot tasks on satellite and aerial images. Moreover they introduced a new counting task to show the object counting ability of such model.

In the context of text generation, transformer architecture became recently the state-of-the-art choice for all the tasks involving text. GPT-2, developed by Radford et al., is a powerful and light model for text generation. It predicts the next word in a sentence based on all the previous words, enabling it to generate coherent and contextually relevant text once trained on a large dataset.

Using images to create textual captions is not an easy task and ad-hoc architectures and pipelines are used to deal with this kind of tasks. Mokady et al. proposed a pipeline using a fine-tuned CLIP as backbone, a transformer block appended to the final image embeddings of CLIP that serves to project the CLIP embeddings to the GPT-2 embeddings, as depicted in 1. A similar approach is used by Silva et al., they propose RS-CapRet, a vision and language method specifically designed for remote sensing tasks such as image captioning and text-image retrieval. RS-CapRet utilizes a LLamaV2 [10] model along with a fine-tuned CLIP encoder adapted to remote sensing imagery. By training simple linear layers to bridge the image encoder and language decoder while keeping other parameters frozen, RS-CapRet achieves state-of-the-art or competitive performance.

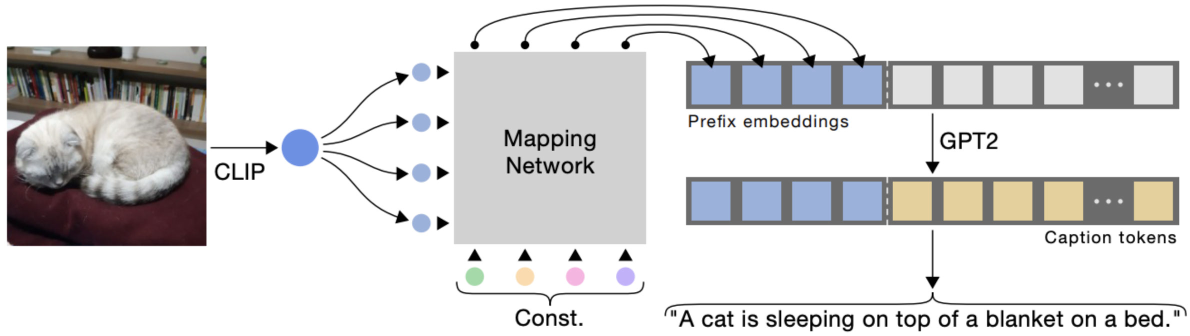

Figure 1: Overview of the pipeline used, in this work the mapping network is a single linear layer that projects the CLIP embeddings to the GPT-2 embedding size, the Const. values are not considered here. The image is taken from [4]

III. Datasets

For this project, the CAP-4 dataset of annotated satellite images specifically designed for this task will be considered. This dataset is a composition of 4 different datasets: NWPU-Captions[1], RSICD-Captions[3], UCM-Captions[5], and Sydney-Captions[5].

1) NWPU-Captions: This dataset contains 31,500 images, each manually annotated with five unique sentences, with a total of 157,500 sentences. These images cover a broad range of 45 different scene classes, making the dataset diverse and comprehensive. The spatial resolution varies from 30 to 0.2 meters/pixel for most of the images. All the images are collected from Google Earth.

2) UCM-Captions: This dataset was constructed based on the UCMD dataset that was originally used for classification tasks [11]. It contains 2,100 images with 21 different categories. The image size is 256 × 256 pixels. Each image is describerd with five different captions with 10,500 descriptions in total. The images comes from aerial orthoimagery with a resolution of one foot per pixel.

3) Sydney-Captions: This dataset was built on top of the Sydney dataset, which was originally used for classification tasks. All the images are derived from a 18,000×14,000 big image of Sydney from Google Earth. The images are 500×500 pixels and belong to seven scene categories, with each image annotated by five different descriptions, totaling 3,065 sentences.

4) RSICD: In this dataset, all the images are collected from different sources and cropped to 224 × 224 pixels with various spatial resolutions. RSICD contains 10,921 images with 30 different categories, and the total number of descriptions is 24,333.

A. Merging datasets

Not all datasets has the same resolution and some do not always have five captions for each image. To solve the first problem I resized the images to a size of 224x224, which is the size that CLIP accepts as input. For the latter, at training time, only one caption is randomly selected, this strategy helps augmenting data because at every epoch the model will receive a different caption for the same sample as input. All of these datasets are already split by design into training, validation, and test sets.

IV. Method

The pipeline proposed here is inspired by the work of Mokady et al. [4] where a fine-tuned version of CLIP is used to calculate the image embedding that are projected to the GPT-2 input token embedding. This is done with the help of a Transformer block that takes in input the CLIP embedding and project to a dimension of $gpt\_embedding * prefix\_lenght$ and then reshaped to create a sequence of embeddings of dimension $[batch\_size, prefix\_lenght, gpt\_embedding]$ as depicted in Fig.1. In my case I had to abstract the different encoders and assume that their output has already a shape $[batch\_size, prefix\_lenght, encoder\_dim]$, in that way I can append a simple linear layer on top of the encoder to project and match the dimension of the GPT-2 decoder. I will detail in the relative subsections how this abstraction is achieved for each encoder.

For all training methods, I used the AdamW optimizer and a cosine scheduler with warmup. The warmup phase helps the AdamW optimizer achieve better convergence properties in the first stages of the training process. While the cosine scheduling gradually decreases the learning rate during the training to mitigate overfitting.

To effectively select the best models during training I decided to keep the model with the highest SPICE score on the validation set. This is because this metric considers the intrinsic meaning of the captions instead that raw words, making it more reliable than the other metrics to evaluate a captioning task in general terms. Using a metric like the BLUE score may lead to select a model that over-fits more on the datasets leading to poor results in a real case scenario.

Evaluation Dataset	Backbone	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	CIDEr	METEOR	SPICE
NWPU	VGG16	0.73	0.57	0.48	0.41	0.61	1.06	0.32	0.24
	fine-tuned CLIP	0.80	0.69	0.61	0.56	0.72	1.42	0.42	0.31
	RemoteCLIP	0.82	0.72	0.65	0.60	0.75	1.53	0.43	**0.35**
	RS-CapRet [8]	**0.87**	**0.79**	**0.71**	**0.66**	**0.78**	**1.92**	**0.44**	0.32
RSICD	VGG16	0.57	0.37	0.26	0.20	0.45	0.55	0.23	0.21
	fine-tuned CLIP	0.62	0.43	0.33	0.26	0.49	0.72	0.26	0.25
	RemoteCLIP	0.68	0.50	0.40	0.32	0.54	0.94	0.30	0.30
	RS-CapRet [8]	**0.74**	**0.62**	**0.53**	**0.46**	**0.65**	**2.60**	**0.38**	**0.48**
UCM	VGG16	0.48	0.28	0.19	0.14	0.39	0.65	0.19	0.18
	fine-tuned CLIP	0.67	0.56	0.50	0.45	0.64	2.21	0.35	0.40
	RemoteCLIP	**0.86**	**0.80**	**0.75**	**0.71**	**0.83**	3.40	**0.48**	**0.54**
	RS-CapRet [8]	0.83	0.76	0.70	0.65	0.79	**3.42**	0.44	0.53
Sydney	VGG16	0.49	0.31	0.24	0.19	0.37	0.71	0.19	0.24
	fine-tuned CLIP	0.55	0.45	0.37	0.30	0.57	0.87	0.32	0.35
	RemoteCLIP	0.57	0.50	0.43	0.37	0.62	0.98	0.38	0.41
	RS-CapRet [8]	**0.78**	**0.69**	**0.61**	**0.54**	**0.70**	**2.40**	**0.38**	**0.43**

Table I: Performance comparison of various models on different datasets using multiple evaluation metrics.

A. *CLIP fine-tuning*

In training the pipeline using CLIP, a two-stage training process is employed. The first stage involves CLIP fine-tuning, and the second focuses on both the decoder and the adapter layer. The objective during CLIP fine-tuning is to maximize the cosine similarity between the embeddings of the two modalities for each sample. I trained the CLIP model for 15 epochs, keeping the model with the lowest validation loss. The learning rate was set to 5e-05, and the batch size was 64. I employed a weight decay of 0.2 and set the second beta of the AdamW optimizer to 0.98. These hyperparameters are the same as those used in the original training of CLIP, with only the learning rate being smaller. The version of CLIP I used is the smaller transformer `ViT-B/32`.

To match the output dimension $[batch\_size, prefix\_lenght, encoder\_dim]$, the linear head on top of the CLIP model, which matches embeddings from different modalities, is discarded. This way the output will be in the form $[batch\_size, num\_patches + 1, encoder\_dim]$ allowing each patch to be considered a token input for the GPT-2 model. The "+1" is the $[CLS]$ token used by the dropped final layer. This approach is preferable to using the final output of CLIP because every patch contains specific information about that part of the image, while the final CLIP embedding contains a more general description of the image. Secondly, if the adapter layer had to project to a dimension of $gpt\_embedding * prefix\_lenght$ it would require way more parameters.

Once the CLIP model is trained, the next step is to fine-tune the GPT-2 model along with the adaptation layer, the second component of the pipeline. For this purpose, I utilized the smaller pretrained GPT-2 model available on HuggingFace. During the training phase, the model is provided with the entire embedded ground truth caption appended to the adapted CLIP embedding. The Hugging Face library automatically shifts the output of GPT-2 to train the model on predicting the subsequent words and returns the correct loss. The training ran for 15 epochs with a learning rate of 1e-05 for the GPT-2 model and 3e-05 for the projection layer. The batch size was 32, with a weight decay of 1e-08 and a dropout rate of 0.2.

B. *RemoteCLIP*

In this case no training is involved and the pre-trained model proposed by Liu et al. is used. The same model dimension, `ViT-B/32`, is used for consistency with the previous method. For the decoder training, the same procedure as the previous step is employed.

C. *VGG encoder*

In this experiment, a VGG16 encoder is used. In particular, only the features coming from the convolutional part of the VGG network are considered. The dimensionality of the output is in the form $[batch\_size, out\_channel, width, height]$ where $out\_channel = 512$ and $width = height = 7$. To coherently transform the dimensionality of this output to $[batch\_size, prefix\_lenght, encoder\_dim]$ I first considered the $out\_channel$ dimension as the $encoder\_dim$. Flattening the last two dimensions, width and height respectively, results in having a set of visual patches, thus $prefix\_lenght = 49$. This approach alignes well to the method used for CLIP: in the 7x7 output, each pixel has 512 channels, which can be seen as an embedding containing high-level features. This provides a similar conceptual representation to CLIP, where each patch is embedded and represented by a high-dimensional vector.

In this case, the training is run end-to-end because this architecture does not allow for the application of any kind of contrastive pre-learning. The same hyperparameters were used as in the previous step, with a learning rate of 1e-05 for the VGG network.

V. Experimental results

After the training of the models, both quantitative and qualitative analyses were performed. All metrics for the various models are reported in Table I. For comparison, the results of RS-CapRet [8] are also included in the table; these results derive from employing the same training strategy with the entire CAP-4 dataset. The data clearly indicate that the most effective model backbone is RemoteCLIP, which outperformed the other backbones. Specifically, for the UCM task, it generally surpassed the performance of the state-of-the-art method, RS-CapRet. It is clear that the least effective backbone model is VGG16.

1) Qualitative analysis: Figures 2, 3, 5 and 4 show some test images, captioned with the three different backbones. It is interesting to see how using RemoteCLIP the counting capabilities emerged from the original paper are not lost: the GPT-2 decoder is able to exploit such capability to output captions with a correct number of objects. This is evident in the second image in Fig. 2 with four planes and in the first two images in Fig. 4, the first showing two freeways and the second two tennis courts. In contrast, the other models struggle to accurately count the number of elements in these images.

In general, captions created using RemoteCLIP well captures and relates the objects present in the image, for example in Fig. 3 in the last image it captures the circular building surrounded by trees, while CLIP sees a bare land that is not present and VGG cannot even describe the main building.

Another interesting observation is that in the last image of Fig. 4, RemoteCLIP accurately finds the narrow road that goes through the area. Interestingly, while none of the ground truth captions mention a road, this is likely due to human tendency to focus on the central elements of an image, often not considering details at the borders. This highlights a key difference between how humans describe images and how RemoteCLIP (but in general any machine learning algorithm) processes them.

The model utilizing the VGG16 backbone sometime fails even to classify the image correctly: it is the case of the first image in Fig. 5 where it confuses the river with a railway station. Even on the second image of Fig. 5 it classifies an industrial area as a residential area.

VI. Conclusions

Different encoders and different training strategies lead to different results. As we have seen, RemoteCLIP is the most effective and ready-to-use foundation model and using it as a decoder for this kind of pipeline leads to very promising results, almost reaching state-of-the-art methods. The most important finding here is that the inner capabilities of RemoteCLIP, such as counting objects are transferred to the entire model. Moreover, from the qualitative analysis emerged that all the models based on CLIP can always correctly find the category the image belongs to, while VGG16 can sometimes misclassify objects and categories in the image.

VII. Future works

To further improve this pipeline the first step to do is to use bigger models, for example using the `ViT-L/14` version of RemoteCLIP and a more complex decoder such as a LLamaV2. With the limited amount of resources training such big models was infeasible for this work. Other techniques may be used such integrate in the pipeline a visual grounding element to further enhance the precision of the encoder.

References

[1] Qimin Cheng, Haiyan Huang, Yuan Xu, Yuzhuo Zhou, Huanying Li, and Zhongyuan Wang. Nwpu-captions dataset and mlca-net for remote sensing image captioning. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022. doi: 10.1109/TGRS.2022.3201474.

[2] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. Remoteclip: A vision language foundation model for remote sensing, 2024.

[3] Xiaoqiang Lu, Binqiang Wang, Xiangtao Zheng, and Xuelong Li. Exploring models and data for remote sensing image caption generation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2183–2195. doi: 10.1109/TGRS.2017.2776321.

[4] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clipcap: Clip prefix for image captioning, 2021.

[5] Bo Qu, Xuelong Li, Dacheng Tao, and Xiaoqiang Lu. Deep semantic understanding of high resolution remote sensing image. *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5, 2016. URL https://api.semanticscholar.org/CorpusID:16072431.

[6] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[8] João Daniel Silva, João Magalhães, Devis Tuia, and Bruno Martins. Large language models for captioning and retrieving remote sensing images, 2024.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[11] 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*, GIS '10, page 270–279, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450304283. doi: 10.1145/1869790.1869829. URL https://doi.org/10.1145/1869790.1869829.

Table II: Occurrence of the most occurring captions in the datasets

	NWPU	RSICD	Sydney	UCM
Sentences from NWPU-Captions dataset, containing 157,500 captions				
'This is a dense forest'	415 (0.003%)	2	0	8
'The entire image is dominated by grass'	435 (0.003%)	0	0	0
'The snow berg is consist of bare land and white snow'	448 (0.003%)	0	0	0
'The bare and green terrace is next to some trees'	290 (0.002%)	0	0	0
Sentences from RSICD-Captions dataset, containing 24,333 captions				
'many buildings and green trees are in a dense residential area'	0	592 (0.024%)	0	0
'many pieces of farmlands are together'	0	434 (0.017%)	0	0
'many buildings are in an industrial area'	0	292 (0.012%)	0	0
'it is a piece of green meadow'	0	218 (0.009%)	0	0
Sentences from UCM-Captions dataset, containing 10,500 captions				
'There is a piece of farmland'	0	0	0	111 (0.011%)
'There is a piece of cropland'	0	0	0	111 (0.011%)
'It is a piece of farmland'	0	0	0	111 (0.011%)
'It is a piece of cropland.	0	0	0	111 (0.011%)
Sentences from Sydney-Captions dataset, containing 3,065 captions				
'a residential area with houses arranged neatly and some roads go through this area'	0	0	79 (0.026%)	0
'a residential area with houses arranged neatly while many plants on the roadside'	0	0	59 (0.019%)	0
'this is a part of deep green sparkling sea'	0	0	165 (0.054%)	0
'a part of ocean with deep green waters'	0	0	56 (0.018%)	0

Ground Truth	The harbor has lots of neatly docked boats and some buildings are next to the harbor	Four planes of different sizes were on the marked ground	Chaparral is evenly distributed on the sand	There are many rectangular farmlands of different sizes and shades
VGG16	The harbor has lots of neatly docked boats and the color of the waters is green	Three planes of different sizes are parked on the airport	Some disconnected chaparrals of different sizes grow in the desert	There are some green rectangular farmlands of different colors
Fine-tuned CLIP	The harbor has lots of neatly docked boats and some buildings are next to the harbor	Two planes are in the parking lot next to the building	Chaparral is evenly distributed on the sand	Many dense, neatly arranged dark green, light green, Khaki and brown mixed rectangular farmlands of different sizes
Remote CLIP	The harbor has lots of neatly docked boats and the color of the waters is green	Four planes parked in a line on the airport and the lawn behind them	Some disconnected chaparrals grow in the desert	Many dense, neatly arranged dark green, light green, Khaki and brown mixed rectangular farmlands of different sizes

Figure 2: Results on some of the NPWU samples

Ground Truth	several planes are parked in an airport near several buildings with a parking lot	several green trees are around a baseball field .	a circle gray center building is surrounded by some green trees and a circle road with several cars
VGG16	Several planes are parked in an airport near some buildings and green trees	A baseball field is near several green trees and several buildings	Many green trees and several buildings are near a viaduct with many cars
Fine-tuned CLIP	Many planes are in an airport near some buildings	There are two baseball diamonds in the grass surrounded by the houses and the forests	The palace is on the bare land next to some trees
Remote CLIP	Many planes are parked near a terminal in an airport	A baseball field is near several green trees and a building	Some green trees are around a circle building

Figure 3: Results on some of the RSICD samples

Ground Truth	Two straight freeways parallel forward with some cars on them	There are two tennis courts surrounded by some trees	A medium residential area with houses surrounded by lawn
VGG16	There are some green trees beside the overpass	The tennis court is on the grass next to some buildings and trees	Many buildings and green trees are in a medium residential area
Fine-tuned CLIP	The freeway goes through the forest with some cars	The tennis court is on the bare land next to some trees	There are many roads and neatly arranged houses and trees and large lawns in densely populated areas
Remote CLIP	There are two straight freeways closed to each other with some plants beside them	There are two tennis courts arranged neatly and surrounded by some trees	This is a medium residential area with a narrow road goes through this area

Figure 4: Results on some of the UCM samples

			
Ground Truth	there are some white boats on the river with some houses on the bank	an industrial area with many white buildings arranged densely and some containers beside	a part of ocean with deep green waters
VGG16	Many buildings and green trees are in two sides of a railway station	Many buildings and some green trees are in a dense residential area	This is a part of deep green sparkling sea
Fine-tuned CLIP	There are many buildings of different shapes beside the beach	In an industrial area, There are neatly planned factories and roads, Green belts, And parking lots for cars and trucks	This is a part of deep green sparkling sea
RemoteCLIP	A wide river with deep green waters go through a residential area with some houses beside the river	An industrial area with many white buildings and some roads go through this area	This is a part of deep green sparkling sea with a white line in the middle

Figure 5: Results on some of the Sydney samples