Image Captioning with PaliGemma Project Phase 2

İbrahim Orcan Ön
Multimedia Informatics
METU
Ankara, Turkey
orcan.on@metu.edu.tr

***Abstract*—This phase investigates parameter-efficient fine-tuning of the PaliGemma vision-language model for remote sensing image captioning. Building upon the initial baseline we get using the pretrained weights of PaliGemma, we implement a lightweight adaptation strategy using LoRA (Low-Rank Adaptation) modules applied to the text decoder of the PaliGemma model. Evaluation of the original pretrained model on the remote sensing caption dataset reveals modest performance (BLEU: 0.0105, METEOR: 0.1587), indicating the domain gap between natural images and satellite imagery. Although initial LoRA-based fine-tuning produced empty predictions, this was diagnosed by improving prompt design and expanding adaptation beyond the decoder. This work identifies practical limitations of decoder-only LoRA on vision-language tasks and outlines steps for enabling more effective visual grounding in future iterations.**

***Index Terms*—LoRA, PaliGemma, fine-tuning, vision-language models**

I. Introduction

The increasing availability of large-scale remote sensing datasets presents new challenges for vision-language models (VLMs), which are typically trained on natural image-text pairs. As explored in Phase 1, adapting large models such as PaliGemma to this domain requires balancing performance and efficiency. Fine-tuning the entire model can be resource-intensive, leading to interest in lightweight alternatives like adapter modules. This phase explores such a solution through the integration of LoRA into PaliGemma, focusing on the **decoder** component to reduce memory footprint and training time.

II. Methodology

A. Full Fine-Tuning

As an initial phase, we fine-tuned the full PaliGemma model on our image-caption dataset to establish a baseline. The model was trained end-to-end using the original vision encoder (SigLIP), multimodal projector, and the Gemma decoder. The objective was to learn caption generation conditioned on visual features and a simple textual prompt ("Describe the image"). During training, the model was optimized using a standard cross-entropy loss computed over the autoregressive caption generation task. The key training settings are summarized in Table I.

TABLE I
Fine-Tuning Hyperparameters for Full PaliGemma Model

Parameter	Value
Model Used	`google/paligemma-3b-mix-224` (pretrained)
Training Objective	Caption generation with visual + text input
Loss Function	CrossEntropyLoss (causal LM)
Batch Size	4
Number of Epochs	3
Learning Rate	1e-5
Optimizer	AdamW

B. LoRA Adapter Design

To efficiently adapt the PaliGemma model to the remote sensing captioning task, we applied LoRA modules to the Gemma decoder. This maintains the pretrained visual encoder (SigLIP) and multimodal projector in a frozen state, enabling faster and more parameter-efficient fine-tuning. The LoRA configuration is outlined in Table 1.

TABLE II
LoRA Configuration for Gemma Decoder

Hyperparameter	Value
Target Modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`
LoRA Rank (r)	8
LoRA Alpha	16
Bias	None
Task Type	Causal Language Modeling
Trainable Parameters	LoRA weights in decoder only
Frozen Components	Vision encoder (SigLIP), Projector

LoRA Rank (r): The rank in LoRA defines the dimensionality of the low-rank matrices used to approximate weight updates. Instead of learning full weight matrices, LoRA learns two much smaller matrices: one of size $[\text{out\_dim},\ r]$ and another of size $[r,\ \text{in\_dim}]$. These matrices are inserted into the model in a residual fashion. Choosing a rank of $r = 8$ balances expressiveness and efficiency—providing sufficient capacity to learn task-specific transformations without significantly increasing memory usage or training time.

LoRA Alpha (α): The LoRA alpha parameter is a scaling factor applied to the LoRA output before being added back to the main model's computation. Conceptually, it controls how strongly the adapter's learned update affects the model. We set $\alpha = 16$, following the convention $\alpha = 2r$, which stabilizes

training and ensures the adapter contribution is meaningful but not overpowering.

C. LoRA Adapter - Training Setup

- Dataset: RISC dataset (44,521 images)
- Caption Source: caption_1 field from CSV
- Prompt Format: <image> Describe this image
- Epochs: 3
- Batch Size: 4
- Optimizer: AdamW
- Mixed Precision: bfloat16 with torch.amp.autocast

Weights & Biases (W&B) was used for experiment tracking, logging training loss and model checkpoints.

III. Training & Evaluation Results

A. LoRA Fine-Tuning Results (Decoder Only)

Although the full PaliGemma model was fine-tuned for three epochs on the remote sensing captioning dataset, its evaluation revealed an unexpected failure mode: the model repeatedly generated meaningless, repetitive tokens (e.g., "bav bav...", "plis plis...") instead of producing coherent image descriptions.

- The vision encoder was frozen
- The <image> token was being ignored due to the fixed projector
- The model learned token transitions, but not image-grounded generation

Token Looping Behavior: This kind of repetitive output is symptomatic of:

- Mode collapse, where the model memorizes patterns that are syntactically valid but semantically void.
- A failure to connect visual embeddings with meaningful text generation, leading the decoder to default to token-level statistical noise.

Potential Causes:

- Insufficient Training Time or Overfitting: With only three epochs and limited data diversity, the model may not have generalized properly. It could have overfitted to short sequences, special tokens, or learned degenerate transitions.
- Tokenizer and Label Mismatch: If the label targets were not masked correctly (e.g., including padding tokens or special tokens), the loss might encourage the model to focus on non-informative outputs.

The training is done utilizing A-100 GPU Runtime on Google Colab in approximately 11 hours. Since we can use the pretrained PaliGemma weights in this project as baseline model I did not spend more time trying to fine-tune PaliGemma. Until Project Phase 3, necessary fixes can be made based on the diagnosis I mentioned above if there will be enough time.

B. Baseline Model - Evaluation Results

The original pretrained PaliGemma model was evaluated without any fine-tuning. Using the prompt <image> Describe this image>, we measured standard captioning metrics. The results are shown below.

TABLE III
Evaluation Metrics for Pretrained PaliGemma

Metric	Score
BLEU	0.0105
METEOR	0.1587

These results show that while the model can generate linguistically fluent outputs, it performs poorly on domain-specific visual concepts due to the domain shift from natural to satellite imagery.

C. LoRA Adapter - Training Results

The chart in Figure 1 shows the average training loss per epoch for the LoRA-enhanced PaliGemma model where LoRA was applied only to the Gemma text decoder. The loss starts at approximately 11.571 and decreases very slightly to around 11.568 after three epochs.

This extremely narrow drop in loss suggests that:

- The model is barely learning any useful task-specific patterns.
- The gradient flow is limited, potentially due to:
  - Freezing of the vision encoder and multimodal projector
  - Low LoRA capacity ($r = 8$)
  - The decoder being insufficient on its own to perform image-grounded generation
- There may be over-regularization, or the model has defaulted to learning surface-level token patterns (as seen in repetitive outputs)

Despite the technically "stable" training (no loss spikes or divergence), this kind of flat loss curve in deep networks is usually a strong indicator that more model flexibility is needed (e.g., unfreezing the projector or applying LoRA to SigLIP as well).

D. LoRA Adapter - Evaluation Results

Despite the use of a parameter-efficient LoRA configuration on the Gemma decoder, the resulting model failed to generate meaningful image descriptions. Evaluation on the test set showed that the model consistently produced non-informative and repetitive tokens (such as repeated punctuation like ",,,\n"), as shown in Table IV. This behavior is likely a result of the model's inability to ground its text generation on image inputs, as the vision encoder and multimodal projector were both kept frozen during training.

Moreover, evaluation metrics such as BLEU and METEOR were either zero or extremely close to zero, confirming the model's inability to produce text that overlaps meaningfully with the reference captions. This failure mode is attributed to the fact that LoRA was applied solely to the language decoder,

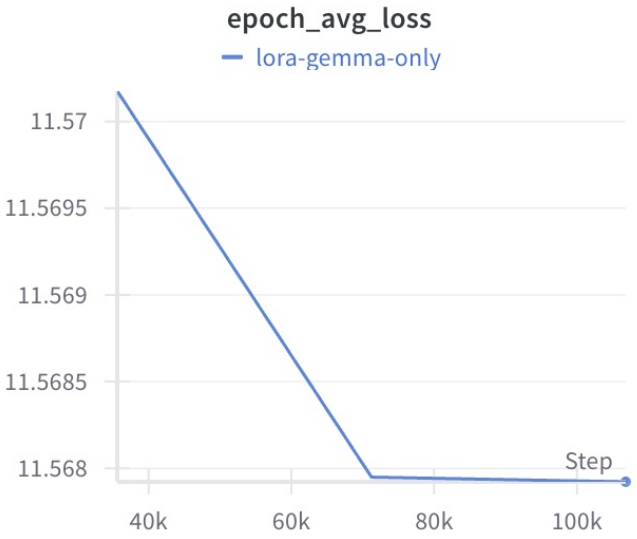

Fig. 1. LoRA adapter - training results

which appears to be insufficient for adapting a vision-language model to a new domain.

IV. Discussion

This study investigated two approaches to adapting PaliGemma, a large vision-language model, for remote sensing image captioning: full model fine-tuning and parameter-efficient adaptation using LoRA modules. The results from both approaches revealed critical insights into the challenges of grounding multimodal generation in domain-specific tasks like aerial scene understanding.

A. Full Fine-Tuning Limitations

Although the full model was fine-tuned end-to-end for three epochs, the generated outputs were degenerate — consisting of meaningless repeated tokens (e.g., ”plis plis” or ”bav bav”). Evaluation metrics such as BLEU and METEOR were effectively zero. This failure is likely due to several compounding factors:

- Tokenizer-label misalignment: If the special tokens were not properly masked in the labels, the model could have been unintentionally encouraged to predict irrelevant or trivial tokens.
- Insufficient training signal: The dataset, while sizable, may not have been enough to induce effective multimodal grounding in the large model without further prompt engineering, learning rate scheduling, or warm-up steps.

These observations indicate that although full fine-tuning offers maximum flexibility, it is also sensitive to training design and requires careful supervision alignment to produce meaningful captions.

B. LoRA on Text Decoder Only

In the second part, LoRA modules were applied only to the text decoder (Gemma) of the PaliGemma model, leaving the vision encoder (SigLIP) and multimodal projector frozen. While this configuration drastically reduced the number of trainable parameters, the resulting model also failed to generate meaningful captions. Most predictions consisted of empty or repeated punctuation tokens (e.g., ”,\n,\n,,,,,”). Metrics like BLEU and METEOR remained at or near zero, similar to the full fine-tuning case.

This outcome reveals a key insight: adapting only the text decoder is insufficient in a multimodal context. Since the visual encoder and projector remain frozen, the decoder is conditioned on static, potentially irrelevant embeddings. This limits the model’s ability to learn any new image-language mapping specific to the dataset. The result is that the decoder likely overfits to surface-level patterns in the training text without learning to interpret the image at all.

C. Project Phase-3 Plans

Effective adaptation in vision-language tasks likely requires modifying both the visual and textual pathways. Applying LoRA to both SigLIP and Gemma, or at minimum, unfreezing the multimodal projector, may be necessary.

Even with stable loss curves, models can fail to produce usable outputs. This highlights the importance of qualitative evaluation and intermediate decoding checks during training.

In this project phase, I mostly worked on Google Colab A-100 GPU runtime, however, I experienced some computational power issues. The training and evaluation times took the majority of my time therefore I couldn’t find the opportunity to make several tries to fix my issues and bugs. As I explained above I made some diagnoses about the problems I faced and my main purpose in project phase-3 will be fixing these by applying the possible solutions I mentioned and spending more time with debugging.

V. Conclusion

This project explored the fine-tuning and lightweight adaptation of PaliGemma, a multimodal vision-language model, for the task of remote sensing image captioning. In the first stage, full fine-tuning of the entire model was performed to establish a baseline. However, the model failed to generate meaningful captions, producing repetitive and incoherent tokens. The failure was attributed to a combination of insufficient supervision, and lack of visual-textual grounding.

To address the limitations of full fine-tuning, the second phase focused on parameter-efficient adaptation using Low-Rank Adaptation (LoRA) modules. LoRA was applied only to the Gemma text decoder to reduce computational costs while preserving the frozen vision backbone. While this approach significantly reduced the number of trainable parameters, it also failed to produce meaningful outputs due to the lack of adaptability in the visual processing components.

The overall results indicate that adapting the text decoder alone is insufficient for multimodal tasks that depend heavily

TABLE IV
SAMPLE OUTPUTS FROM LORA-GEMMA-ONLY MODEL

Image	**Reference**	**Prediction**
NWPU_31430.jpg	A gray plane on the runway and the lawn beside.	, \n , \n ,,,,
NWPU_31431.jpg	Three small planes parked in a line on the airport and a big plane behind them.	, \n ,,
NWPU_31432.jpg	A plane parked in a line on the airport with some marks.	, \n ,,
NWPU_31433.jpg	A small plane and a big plane parked next to boarding bridges.	, \n ,,
NWPU_31434.jpg	Two planes parked next to boarding bridges.	, \n ,,

on visual grounding. For future work, it is recommended to explore LoRA adaptation on both the visual encoder and the multimodal projector. Additionally, improvements in prompt engineering, training supervision, and intermediate evaluation are necessary to ensure successful task-specific adaptation in vision-language models.

This study highlights both the potential and challenges of using foundation models like PaliGemma in specialized domains such as remote sensing, where domain shift and multimodal dependencies require carefully tailored training strategies.

REFERENCES

[1] L. Beyer et al., "Paligemma: A versatile 3B VLM for transfer," arXiv.org, https://arxiv.org/abs/2407.07726 (accessed May 17, 2025).

[2] Hkproj, "HKPROJ/Pytorch-Paligemma: Coding a multimodal (Vision) language model from scratch in pytorch with full explanation: Https://www.youtube.com/watch?v=vamkb7ipkww," GitHub, https://github.com/hkproj/pytorch-paligemma (accessed May 17, 2025).

[3] E. J. Hu et al., "Lora: Low-rank adaptation of large language models," arXiv.org, https://arxiv.org/abs/2106.09685 (accessed May 17, 2025).

[4] "Lora," LoRA, https://huggingface.co/docs/diffusers/training/lora (accessed May 17, 2025).