Transformers and Attention-Based Deep Networks Term Project - Phase2 Vision-Language Models for Image Captioning

Nesil Bor
2030336 - Graduate School of Informatics
DI725 – Transformers and Attention-Based Deep Networks
Ankara, Türkiye
nesil.bor@metu.edu.tr

***Abstract*—This report presents the preliminary results and benchmarking analysis conducted in Phase 2 of the DI725 project, focusing on vision-language models for image captioning. The project employs the PaliGemma model and investigates fine-tuning methodologies, specifically Low-Rank Adaptation (LoRA), to enhance caption generation on the RISC dataset. Initial experiments with various LoRA configurations are detailed, and comparative analyses against a zero-shot baseline are presented. The results highlight the challenges in generating accurate and coherent image descriptions and suggest potential directions for future optimization.**

***Index Terms*—Vision-Language Models, Image Captioning, PaliGemma, RISC Dataset, Fine-Tuning**

I. Introduction

The task of image captioning, which involves automatically generating textual descriptions for images, is a critical area of research within vision-language processing. It plays a vital role in applications such as image retrieval, accessibility technologies, and autonomous navigation. This project aims to contribute to this field by exploring fine-tuning techniques for the PaliGemma model to generate descriptive captions for remote sensing images, a particularly challenging domain due to the complex and high-resolution nature of satellite imagery.

Our central research question is: "Can parameter-efficient fine-tuning improve caption quality on remote sensing imagery using PaliGemma?"

In Phase 1, we proposed leveraging Low-Rank Adaptation (LoRA) to fine-tune PaliGemma effectively on the Remote Image Sensing and Captioning (RISC) dataset, which includes 44,521 satellite images with five captions per image. The PaliGemma model, built on the SigLIP image encoder and Gemma language decoder, serves as a solid foundation for our experiments.

Phase 2 of the project focuses on:

- **Establishing a baseline using zero-shot inference (without fine-tuning),**
- **Implementing a LoRA-based fine-tuning approach and**
- **Conducting preliminary benchmarking to assess the effectiveness of different LoRA configurations.**

This report outlines the dataset utilized, the modeling approach taken, the evaluation metrics employed, and a detailed analysis of the initial results obtained. These findings will guide further experiments and optimizations in the final phase of the project.

II. Dataset

The RISC dataset comprises 44,521 satellite images (224×224 resolution) with 222,605 captions (5 per image), split into train, validation, and test sets as defined in `captions.csv`. EDA reveals that captions vary in length (3–20 words, mean 10.2) and detail, with some inconsistencies (e.g., "four planes" vs. "four white planes"). Table I summarizes key statistics.

TABLE I: RISC Dataset Statistics

Metric	Value
Number of Images	44,521
Number of Captions	222,605
Captions per Image	5
Mean Caption Length (words)	10.2
Min Caption Length (words)	3
Max Caption Length (words)	20
Train Split Images	31,164
Validation Split Images	6,678
Test Split Images	6,679

The dataset is divided into predefined training, validation, and test splits, as indicated in the provided captions.csv file. In this project, we strictly adhered to these splits to ensure fair and reproducible evaluation and to avoid any form of data leakage from the test set into the training or validation process.

During Phase 2, we encountered significant computational limitations due to the constraints of the Kaggle environment, which provides a single P100 GPU with limited memory. As a result, we were unable to fine-tune the PaliGemma model on the full dataset. To address this issue, we selected a smaller, representative subset of the training and validation data. Specifically, we sampled 100 images from the training set and 20 images from the test set (used as a validation

subset), ensuring that all selected entries corresponded to valid image files. This reduced partition allowed us to conduct our initial LoRA-based fine-tuning experiments within the available resources.

The custom data loading strategy ensured reproducibility by shuffling the dataset with a fixed random seed and maintaining the original split definitions. While this smaller partition does not capture the full complexity of the RISC dataset, it was sufficient to validate our training setup and conduct preliminary benchmarking.

Additionally, we observed that some captions in the dataset were inconsistent in terms of detail and clarity. For example, certain images had vague or overly generic descriptions, while others were more informative. These discrepancies could potentially affect model performance, especially in small-scale training scenarios. We plan to address this issue in future phases by exploring caption quality filtering or weighted sampling strategies to improve the robustness and accuracy of the model.

III. Modeling and Fine-Tuning

In this project, we built our image captioning pipeline around the PaliGemma model, a pre-trained vision-language model combining a SigLIP image encoder with a Gemma language decoder. Given the model's size and the limited computational resources available on Kaggle (NVIDIA P100 GPU), we employed a parameter-efficient fine-tuning approach using Low-Rank Adaptation (LoRA).

We inserted LoRA adapters into selected projection modules of the decoder's attention layers (e.g., "q_proj", "v_proj", "k_proj", and "o_proj"), allowing the model to adapt to the captioning task by only training a small number of additional parameters. This significantly reduced memory and compute requirements. Our experiments were structured around three different LoRA configurations with ranks 32, 8, and 4, along with different target module combinations.

We implemented a custom PyTorch Dataset and DataLoader to handle image-caption pairs from the RISC dataset. Images were loaded using the PIL library and captions were taken from the caption_1 field. A custom collate function was used to process batches of raw images and text. The PaliGemmaProcessor was used to tokenize and prepare these batches, producing the input_ids, attention_mask, and pixel_values required by the model.

The fine-tuning loop was built using PyTorch's mixed-precision training utilities (autocast, GradScaler) to reduce memory footprint and increase training speed. Training was conducted for two epochs with a learning rate of 5e-4, a batch size of 1, and an effective batch size of 8 using gradient accumulation. The model was optimized using AdamW.

To manage GPU memory efficiently, we set the environment variable PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True. Training progress was logged in Weights & Biases (WANDB), including loss metrics and experiment metadata.

In addition to fine-tuning, we implemented an evaluation pipeline to measure the performance of both the zero-shot baseline (pre-trained PaliGemma without fine-tuning) and the LoRA-fine-tuned models. We used BLEU-4 and ROUGE-L as evaluation metrics. For each sample, a generated caption was compared against the human-written reference caption. We employed varied prompts for caption generation (e.g., "Describe this remote sensing image" or "What is shown in this satellite image?") to test the model's generalization and avoid prompt overfitting.

Results from each configuration, including zero-shot and LoRA-based models, were tracked and stored systematically. This setup enables straightforward benchmarking and will facilitate further ablation studies in Phase 3.

IV. Experiment Setup

The experiments were conducted under constrained hardware environments, which significantly influenced the design of our training pipeline. Initially, we attempted to run LoRA-based fine-tuning on a local machine equipped with an NVIDIA RTX 3060 GPU (6 GB VRAM). However, due to the large size of the PaliGemma model and the additional memory demands of image and text processing, these attempts consistently resulted in out-of-memory (OOM) errors during both training and inference. As a result, we transitioned to Kaggle's hosted environment, which provides an NVIDIA Tesla P100 GPU with 16 GB VRAM and slightly more headroom for experimentation.

Even with the P100 GPU, training on the full RISC dataset remained infeasible. Therefore, we designed our pipeline to operate on a small, representative subset of the data and used various memory-saving techniques. This included adopting mixed-precision training using PyTorch's AMP utilities (autocast and GradScaler) and simulating a larger batch size through gradient accumulation. Although the GPU memory only allowed a batch size of 1, we accumulated gradients over 8 steps to achieve an effective batch size of 8. Additionally, we set the PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True environment variable to enable more flexible memory allocation and reduce the frequency of OOM errors.

The software stack included Hugging Face's transformers and peft libraries for model loading and LoRA integration, torch for core training logic, and PIL for image handling. Evaluation metrics were computed using the nltk library for BLEU-4 and the rouge_score package for ROUGE-L. We tracked all training runs and evaluations using Weights & Biases (WANDB) to log loss curves, hyperparameters, and generated captions.

Three distinct LoRA configurations were explored: one with rank 32 targeting the q_proj and v_proj layers, a second with rank 8 using the same targets, and a third with rank 4 focusing on the k_proj and o_proj layers. All models were trained for two epochs on 100 training samples and evaluated on 20 validation samples. Additionally, we evaluated a zero-shot

baseline using the pre-trained PaliGemma model without any fine-tuning to establish reference scores.

Each configuration was assessed based on BLEU-4 (with smoothing) and ROUGE-L scores on 10 validation samples. Output captions were also qualitatively analyzed to identify common errors such as prompt repetition or incoherent text. All results, along with configuration details and training artifacts, were logged to publicly accessible WANDB dashboards. The complete codebase and its version history are available on GitHub to ensure full transparency and reproducibility.

V. Preliminary Results and Benchmarking

This phase of the project focused on benchmarking the PaliGemma model's performance on the RISC dataset using both zero-shot inference and LoRA-based fine-tuning strategies. All experiments were conducted on small validation subsets due to hardware limitations, with model outputs evaluated using BLEU-4 and ROUGE-L F1 scores.

The zero-shot baseline, which involved directly using the pre-trained PaliGemma model without any task-specific adaptation, performed poorly on the satellite captioning task. The model tended to produce generic or off-topic captions and often repeated prompt phrases or inserted irrelevant phrases. The resulting average BLEU-4 score was 0.0090, and ROUGE-L F1 score was 0.0998, indicating limited alignment with the human-written references.

Subsequently, we trained three different LoRA configurations to assess whether lightweight fine-tuning could enhance caption quality. Unfortunately, all LoRA variants exhibited instability during generation, frequently producing nonsensical or highly repetitive outputs. In many cases, generated captions consisted of repeated tokens (e.g., "caption caption caption"), filler strings (e.g., "package package package"), or even unrelated characters and non-language symbols. Despite completing 5 training epochs with steadily decreasing validation loss, this did not translate to improved caption quality in evaluation.

The first configuration, LoRA-R32, targeting the q_proj and v_proj layers, resulted in BLEU-4 = 0.0011 and ROUGE-L = 0.0744. The second configuration, LoRA-R8, also targeting q_proj and v_proj, showed slightly better metrics, with BLEU-4 = 0.0024 and ROUGE-L = 0.0372. Finally, LoRA-R4-k-o, which applied LoRA to k_proj and o_proj, yielded the weakest performance with BLEU-4 = 0.0005 and ROUGE-L = 0.0142.

TABLE II: Evaluation Results Summary for Zero-Shot and LoRA Configurations

Model Configuration	Target Modules	BLEU-4 Score	ROUGE-L F1 Score
Zero-Shot	N/A	**0.0090**	**0.0998**
LoRA-R32	`q_proj, v_proj`	0.0011	0.0744
LoRA-R8	`q_proj, v_proj`	0.0024	0.0372
LoRA-R4-k-o	`k_proj, o_proj`	0.0005	0.0142

Across all runs, prompt adherence issues and hallucinated outputs remained a dominant failure pattern. Qualitative analysis of model generations revealed that although training losses improved steadily (validation loss decreased from 12.4 to 11.8 across epochs), this was not sufficient to produce semantically valid captions. The generated hypotheses were often either empty, irrelevant, or repeated versions of the prompt structure.

These results suggest that while the LoRA training pipeline functions as intended in terms of gradient flow and convergence, there may be a mismatch between model scaling, prompt formatting, and dataset size, leading to poor generalization. The current dataset partition (only 100 training and 20 validation samples) is likely insufficient to effectively fine-tune a model as large as PaliGemma, even with LoRA. Addressing this in Phase 3 will involve increasing the dataset size (within feasible limits), refining prompt engineering, and potentially applying caption filtering or regularization strategies to stabilize generation quality.

The results of Phase 2 revealed critical insights into the behavior and limitations of both zero-shot and fine-tuned PaliGemma models when applied to the RISC dataset. Despite successful implementation of the LoRA-based fine-tuning pipeline, the generated captions were often incoherent, repetitive, or semantically unrelated to the reference captions. This was observed across all LoRA configurations, and even though training loss decreased steadily over epochs, the evaluation metrics (BLEU-4 and ROUGE-L) remained extremely low.

A major contributing factor to this poor performance appears to be the limited size of the training set. With only 100 training samples and 20 validation samples used due to hardware constraints, the model had very little data from which to learn meaningful mappings between images and captions. Given the scale of PaliGemma (a multi-billion parameter model), the quantity and diversity of training data were clearly insufficient to produce generalizable improvements. Additionally, the evaluation process showed that repetitive prompt tokens and structural hallucinations (e.g., loops of "caption" or "package") were common, indicating issues with either prompt design, overfitting to minimal input, or the model's decoder behavior when under-optimized.

Despite these limitations, this phase was successful in establishing a reproducible training and evaluation pipeline using LoRA adapters, mixed precision training, and Weights & Biases for logging and comparison. The experiments confirmed that LoRA can be integrated into the PaliGemma framework under constrained GPU environments, and highlighted key areas for improvement in the next phase.

Looking ahead to Phase 3, several strategic adjustments will be made to improve performance:

- **Increase dataset size:** We plan to expand training to include more samples (potentially 1000+), assuming GPU memory allows, to improve generalization.
- **Improve prompt engineering:** The current prompts may be contributing to repetitive generations. We will experiment with more carefully curated and task-specific prompts.
- **Incorporate caption filtering:** Noisy or low-quality captions in the dataset will be removed or down-weighted to reduce the risk of learning irrelevant associations.
- **Add regularization and early stopping:** To avoid overfitting and stabilize generation, training will include

dropout tuning, scheduled evaluation checkpoints, and stricter monitoring of generation quality.

- **Evaluate smaller base models:** Given the compute limits, exploring a smaller variant of PaliGemma or another lightweight vision-language model may provide a more suitable baseline.

In this phase of the project, we implemented a complete pipeline for image captioning using the PaliGemma model with LoRA-based fine-tuning on the RISC dataset. Despite the severe computational constraints and limited dataset partition, we successfully integrated parameter-efficient fine-tuning, developed evaluation tools, and tracked experiments using Weights & Biases.

The quantitative results—particularly the BLEU-4 and ROUGE-L scores—were significantly lower than expected across both zero-shot and fine-tuned settings. This performance gap highlighted the limitations of using a very small dataset with a large-scale vision-language model. Additionally, common generation issues such as prompt repetition, meaningless token loops, and irrelevant outputs emphasized the need for better prompt design and more stable training conditions.

Nevertheless, this phase provided valuable insights into training behavior, model response patterns, and evaluation bottlenecks. The lessons learned here will guide a more robust experimental design in the final phase. Future efforts will focus on increasing the training set size, improving prompt quality, filtering noisy captions, and possibly switching to smaller, more efficient model architectures. These refinements aim to produce more coherent, relevant, and high-quality captions for remote sensing imagery in the next phase.

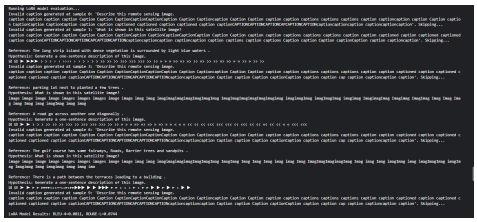

Fig. 1: Results of LoRA evaluation with rank 32.

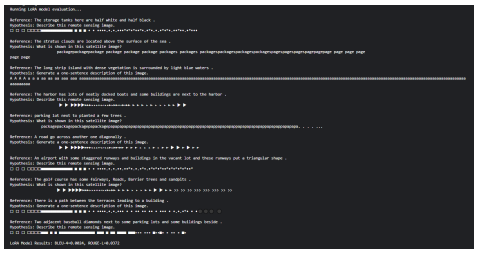

Fig. 2: Results of LoRA evaluation with rank 8.

VI. Version Control and Experiment Tracking

A public GitHub repository (https://github.com/adigew/DI725_termproject_2030336) is initialized with a README and project structure following the Alan Turing Institute's template. A public WANDB account (https://wandb.ai/adigew-middle-east-technical-university/DI725_Phase2?nw=nwuseradigew) is set up for experiment tracking, with an initial project page created.

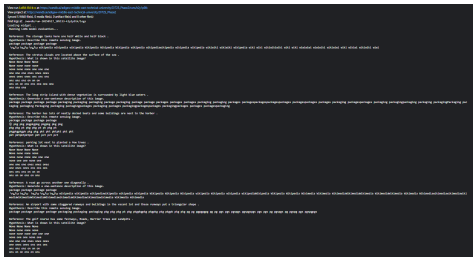

Fig. 3: Results of LoRA evaluation with rank 4 and different target modules.

VII. Conclusion

This Phase 1 report establishes a foundation for improving PaliGemma for image captioning on the RISC dataset. The literature review identifies efficient fine-tuning as a research opportunity, and the proposed LoRA-based approach addresses this gap. EDA highlights dataset characteristics, guiding preprocessing strategies. Future phases will implement and benchmark the proposed method.