OBSS AI Image Captioning Challenge

Participant Report

Name: Pelinsu Kaleli
Email: pelinsu.kaleli@gmail.com
Phone Number: 551 713 4100
Kaggle Username: pelinsukaleli

---

1. Tech Stack

- Python 3.11
- PyTorch 2.6.0+cu124
- Transformers 4.51.3
- Qwen2.5-VL-7B-Instruct - Vision-Language Model
- PEFT (Parameter-Efficient Fine-Tuning) with LoRA
- TRL 0.12.0 - Transformer Reinforcement Learning library
- BitsAndBytes - 4-bit quantization for memory efficiency
- qwen_vl_utils - Qwen Vision-Language utilities
- Google Colab with A100 GPU (training) and T4 GPU (inference)
- Additional libraries: PIL, Pandas, NumPy, tqdm, concurrent.futures

---

2. Summary

This solution implements a fine-tuned Qwen2.5-VL-7B-Instruct model using LoRA (Low-Rank Adaptation) for efficient parameter updates. I have used 4-bit quantization in order to fit the large model in GPU memory for both fine tuning and inference since our model is pretty large given our GPU constraints, while maintaining most of the model's performance. The pipeline processes images through a multimodal transformer that encodes visual and textual information, generating detailed single-sentence captions according to the given prompt. The model was trained on the provided dataset for one full epoch using a prompt that encourages comprehensive, objective descriptions focusing on multiple visual elements, specific attributes, and contextual information.

3. Approach

Model Architecture and Training Strategy

Model Selection: I have tried various models like BLiP, Florence 2, InternVL3-8B and Qwen2.5-VL-7B. Florence 2 performed really well considering its size of just 0.7B parameters, scoring 0.11622 on 30% of the test set, InternVL 3 also performed decently since it also uses Qwen2.5-7B as its language model. In the end I chose the Qwen2.5-VL-7B version as my base model due to its strong performance on our caption tasks among the other models. It is a multimodal transformer that processes both images and text in a unified architecture.

Parameter-Efficient Fine-Tuning: Instead of full fine-tuning, I used Parameter Efficient Fine Tuning due to memory and time concerns. Full fine-tuning a 7B model would take so much time and GPU, therefore this approach was definitely the way to go. I implemented LoRA with the following configuration:

- Rank (r): 8
- Alpha: 16
- Dropout: 0.05
- Target modules: All attention projections (q_proj, k_proj, v_proj, o_proj), MLP layers (gate_proj, up_proj, down_proj), and input/output embeddings (embed_tokens, lm_head)

Quantization Strategy: To handle both the inference and fine tuning of a 7B parameter model on limited GPU memory, I have used:

- 4-bit quantization using BitsAndBytes
- NF4 quantization type with double quantization
- BF16 compute dtype for A100 training, FP16 for T4 inference

Training Configuration:

- Batch size: 1 with gradient accumulation steps: 8 (effective batch size: 8)
- Learning rate: 4e-5 with cosine scheduler
- Warmup ratio: 0.05
- Max gradient norm: 0.3
- Optimizer: Paged AdamW 8-bit
- Training duration: 1 epoch, around 2670 steps (~10 hours on A100)

Inference Optimization

Due to memory constraints and a specific problematic image (350th), I implemented:

- Chunk-based processing (100 images per chunk)
- Aggressive memory clearing after each image
- Intermediate result saving for fault tolerance
- Error handling with generic fallback captions

Prompt Engineering

I developed a comprehensive prompt that guides the model to generate high-quality captions, trying to get similar results of the train set:

```
Unset
Generate a single-sentence, objective, and descriptive caption for the
given image. Strive to be as comprehensive and detailed as possible,
keeping it between 15 to 25 words. Your caption should focus on multiple
significant visible elements: identify primary subjects (e.g., people,
animals) and their actions or notable characteristics; describe key
objects and their attributes; include specific brand names or clearly
legible text from signs/labels; and mention the immediate setting or
context if prominent. The caption must be in the present tense, maintain
a neutral, factual tone (like a museum or news catalog entry), and avoid
subjective opinions.
```

For inference, I slightly modified the prompt to allow 15-30 words and added more specific instructions about spatial positions and composition.

As can be observed from Figure 5.1 (from References), our training loss started very high and after 250 steps it started to plateau around 0.077100, ending at 0.055700. A smaller step size could also be used as the model stopped learning after around 500 to 1000 steps. There could be various reasons as to why, which we will be discussing at the end of this report.

---

4. Sample Outputs

Image: 100011.jpg
Prediction: "A bottle of Pio Cesare wine, featuring a black label with colorful circular designs and elegant typography."
Comment: It is actually a very good caption – correctly identifies the main subject, the branding on the bottle and comments correctly on the design.

Image: 102191.jpg
Prediction: "A desk setup featuring two computer monitors displaying software, a keyboard, and a mouse, alongside a radio equipment display."
Comment: Even though it conveys the image somewhat well, it was not able to correctly identify the roll-up on the right, just commenting it as radio equipment display.

Image: 102314.jpg
Prediction: "A dartboard mounted on a wooden cabinet, featuring Boddingtons branding, surrounded by chalk markings."
Comment: Again a very good caption generated by the model. Gives information about the setting and branding.

Image: 100350.jpg
Prediction: "An image showing various objects and scenes."
Comment: It was one of the failure cases if not the only one. Probably due to its size or resolution the model was not able to generate a caption to it and actually gave me a CUDA out of memory error.

Image: 101739.jpg
Prediction: "A black box labeled ""Hybrid Slinky"" sits on a wooden surface, featuring a shiny, reflective surface and a coin inside."
Comment: The model's understanding of the image is again somewhat true but definitely includes some made up claims as well. It was able to read "Hybrid Slinky" but the "coin inside" part is definitely not true. It misinterpreted the shiny looking thing at the top as a coin and decided that it was inside of the box where it definitely is not.

Strengths: The model performs really well at generating grammatically correct, single-sentence captions that have an objective, descriptive tone which was the main goal of this competition. After performing fine-tuning on the training data, the model was able to focus on multiple visual elements and include specific details like brand names and text but not overdoing it much, something that the base model was doing before fine tuning. Also the use of LoRA allowed efficient adaptation while keeping the base model's general capabilities.

Limitations: It struggles or outright crashes on big images with either too big of a file size or resolution as I have observed from the infamous image 100350.jpg. Another limitation/struggle is that our base model was only a 7B model since we had GPU restrictions. Many SOTA models have 10 to 100x more parameters and definitely perform better than the 7B counterparts. Also I could have used better training settings by trying different learning rates, different LoRA parameters or even seeds as our model did not learn that much after 10% of its training time.

5. References

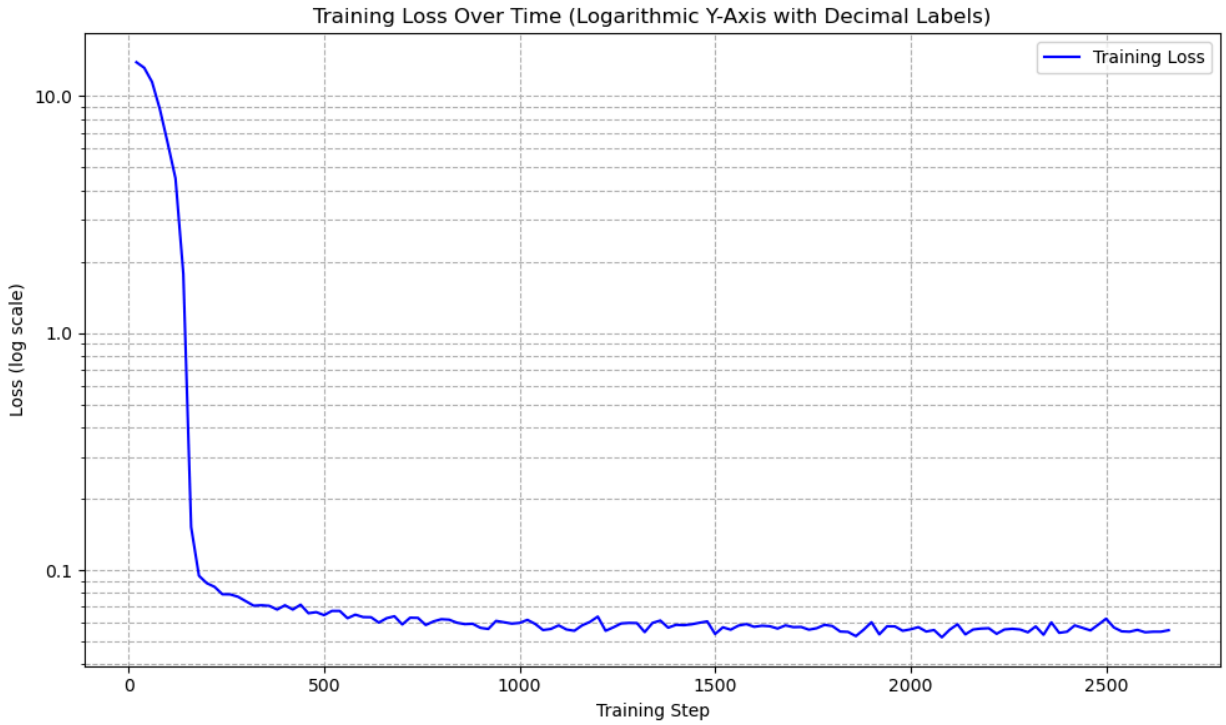

Figure 5.1. - Training Loss Over Time

Image 100011.jpg

Image 102191.jpg

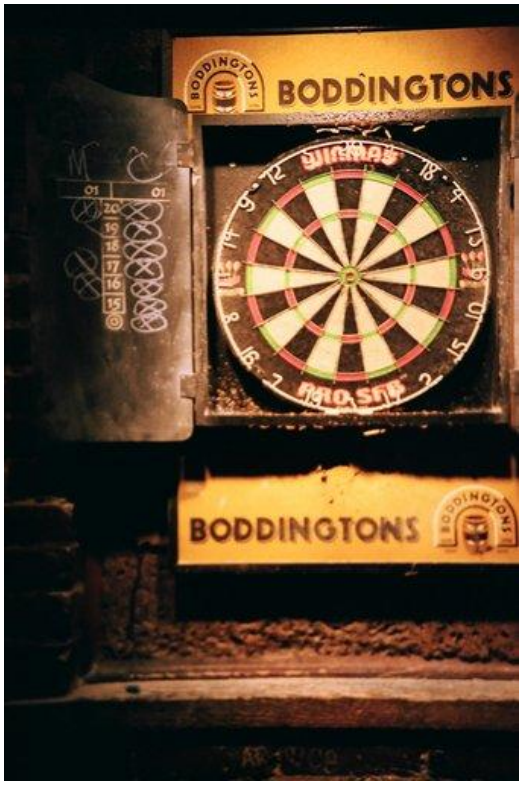

Image 102314.jpg

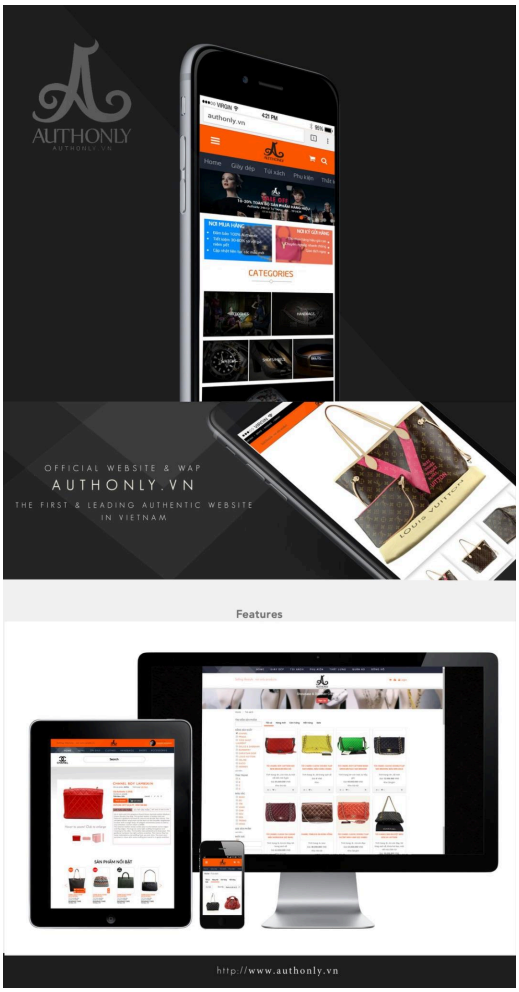

Image 100350.jpg

Image 101739.jpg