Image Captioning for TensorGo

-By V.Kamna

Objective:

The objective of this project is to fine-tune a base language model to generate captions for images, specifically focusing on TensorGo company-related images such as the co-founder, team working, and logo. The model is trained to accurately describe the content of these images in a concise and informative manner. This project demonstrates the steps involved in adapting a pre-trained model for a specific image captioning task, highlighting the improvements before and after fine-tuning.

What the Project is All About:

This project focuses on creating an automated image-captioning system using a pre-trained language model (LLM) that is fine-tuned on a custom dataset related to TensorGo. The dataset contains six images, including the co-founder, team at work, and the company logo. After preprocessing the data and fine-tuning the base model, the system can generate accurate captions for images. The project also includes validation and testing with new images, demonstrating the robustness and flexibility of the fine-tuned model.

Selection Criteria for the Base LLM:

The **microsoft/git-base** model was selected as the base LLM for this project. This model is specifically designed for visual-language tasks and provides a solid foundation for generating captions based on images. The selection was based on:

1. **Pre-trained weights**: It allows leveraging the large corpus of pre-trained image-text associations.
2. **Task compatibility**: The model architecture is well-suited for image captioning.
3. **Flexibility for fine-tuning**: The model can be adapted to various image-related tasks with minimal changes.

Task-Specific Considerations for Fine-Tuning:

The fine-tuning process focused on ensuring that the base LLM could generate relevant and accurate captions specific to the TensorGo company. Several task-specific considerations were made:

Niche Dataset: The custom dataset contained company-specific images (e.g., co-founder, team working, logo). Fine-tuning on this narrow domain ensures the model specializes in generating context-specific captions.

Accuracy in Captions: The captions needed to be accurate, not just general descriptions. The model was tested on how well it could identify unique elements such as logos or team dynamics.

Diverse Testing: After training, additional images were tested, including a grayscale image, to ensure the model could still generate accurate captions in different contexts.

Data Preparation and Preprocessing Steps:

1. **Image Collection**: Images related to the TensorGo company were collected using Google. They included images of the co-founder, the working team, and the logo.
2. **Captioning**: Each image was manually captioned to create a training set for the model. JSON files were created that paired each image with its corresponding caption.
3. **Preprocessing**: The images were resized, normalized, and encoded in the format required by the LLM for fine-tuning.
4. **Validation**: The validation was performed using an image from the training set to ensure the model could recall and caption accurately.

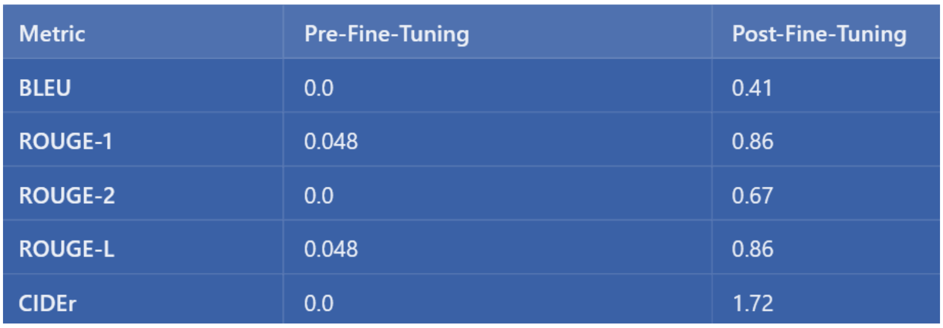
Fine-Tuning Hyperparameters and Optimization Strategies:

1. **Learning Rate**: A small learning rate of 5e-5 was used to ensure the model adjusts slowly, preventing overfitting while retaining its pre-trained knowledge.
2. **Batch Size**: A batch size of 2 was chosen to optimize the balance between memory usage and training speed.
3. **Epochs**: The model was fine-tuned for 50 epochs to allow sufficient updates.
4. **Optimizer**: AdamW optimizer was used, which is well-suited for transformer models and helps in maintaining stable training by adjusting learning rates dynamically.

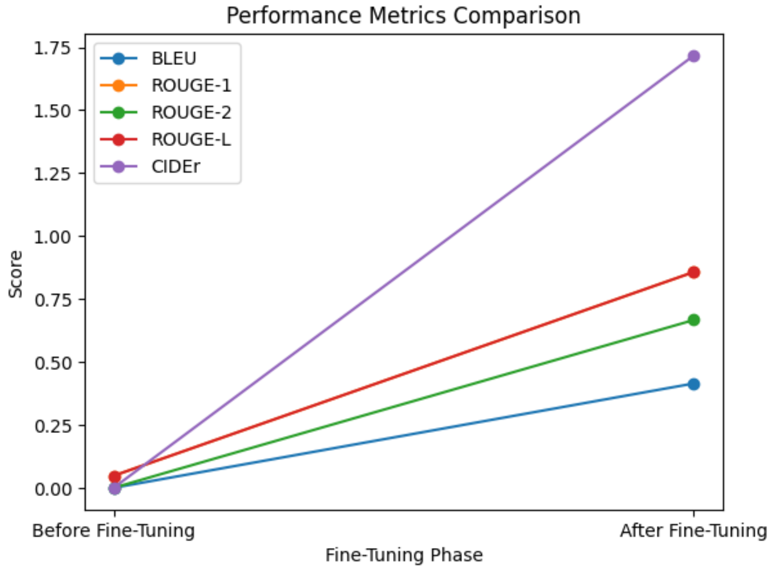
Evaluation Metrics and Performance Analysis:

The fine-tuning results were evaluated using the following metrics:

1. **BLEU (Bilingual Evaluation Understudy Score)**: Measures the precision of generated captions based on reference captions.
2. **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**: Measures the overlap between generated and reference captions.
3. **CIDEr (Consensus-based Image Description Evaluation)**: Measures the consensus of the generated captions with human consensus.



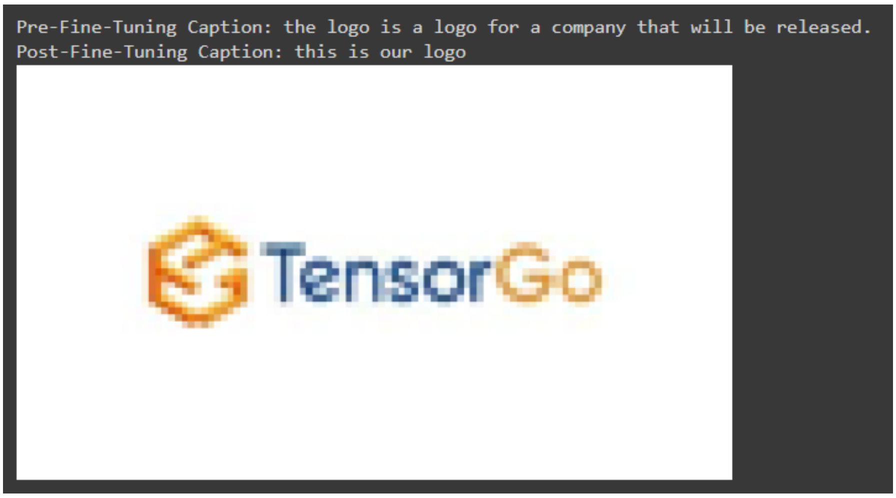
Metric	Pre-Fine-Tuning	Post-Fine-Tuning
BLEU	0.0	0.41
ROUGE-1	0.048	0.86
ROUGE-2	0.0	0.67
ROUGE-L	0.048	0.86
CIDEr	0.0	1.72



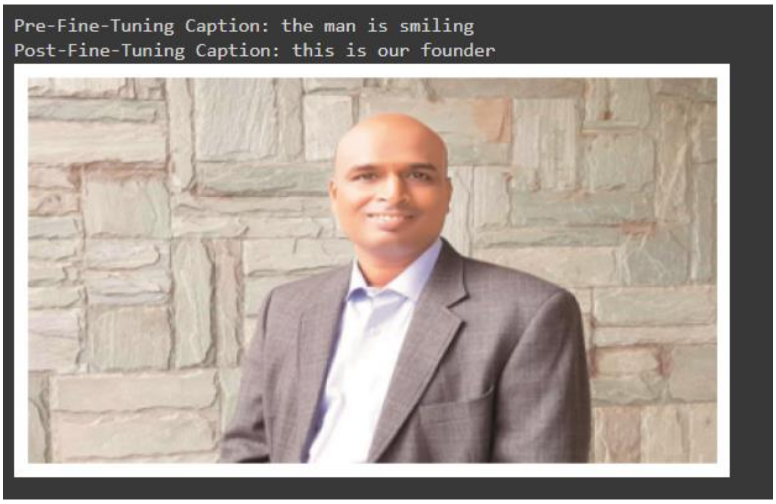
After fine-tuning, the model showed significant improvements in BLEU, ROUGE, and CIDEr scores, demonstrating that it could accurately caption the images specific to the TensorGo dataset.

Test data: Consisted of a diverse set of images that were not included in the training dataset. These images featured different subjects, backgrounds, colour and lighting conditions to assess the model's ability to generalize to unseen data.

COMPARISON OF CAPTIONS BEFORE AND AFTER FINE TUNING ON TEST IMAGES



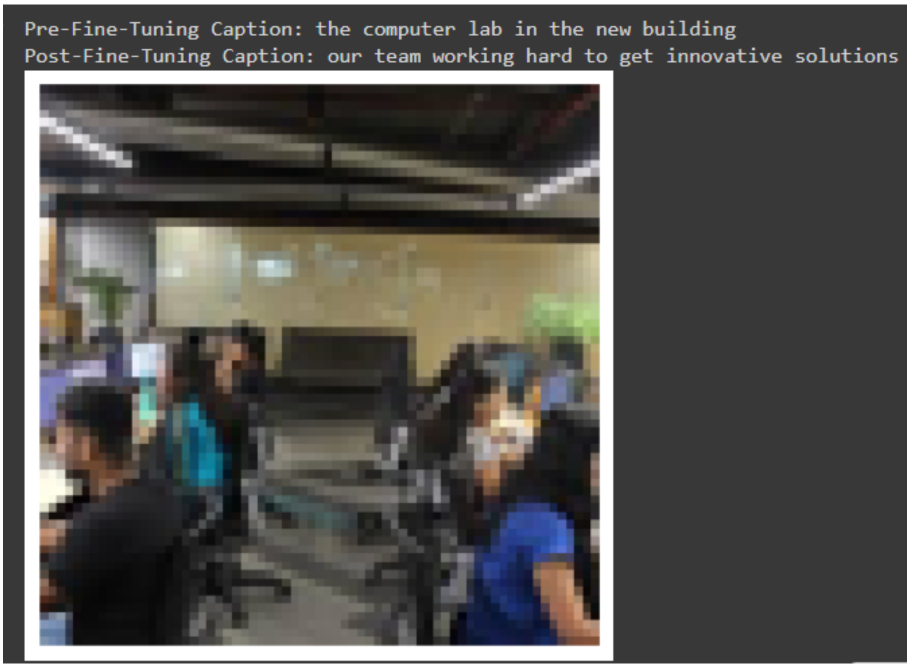
```
Pre-Fine-Tuning Caption: the logo is a logo for a company that will be released.
Post-Fine-Tuning Caption: this is our logo
```



```
Pre-Fine-Tuning Caption: the man is smiling
Post-Fine-Tuning Caption: this is our founder
```

Pre-fine-tuning, the model produced captions that were generic and inaccurate. For example, it described an image of the founder as “the man is smiling.”

Post-fine-tuning, it correctly identified the individual as the founder, generating the caption “this is our founder.” Thus we can say that the fine-tuned model is picking up on important details and providing meaningful and relevant descriptions.



```
Pre-Fine-Tuning Caption: the computer lab in the new building
Post-Fine-Tuning Caption: our team working hard to get innovative solutions
```

CONCLUSION:

This project successfully demonstrates the potential of fine-tuning a pre-trained language model for image captioning tasks. By leveraging the microsoft/git-base model and a custom dataset specific to TensorGo, the system was able to generate accurate and informative captions for company-related images.

APPLICATIONS:

This project can be applied to real-world companies by automating tasks like content generation, branding, and documentation. For instance, companies can use it to automatically generate captions, social media posts, or product descriptions, saving time and improving consistency. This could benefit marketing, customer service, and accessibility.

It can also be used as an assistant for visually impaired people by providing them descriptions.

Other applications include:

Brand-Specific Image Captioning

Product Cataloging in E-Commerce

Medical Imaging – providing descriptions of X-Ray etc.

THANK YOU.