Takeaways from this attempt:

- Need to diversify training data to include images of a diagram
- We want data with more descriptive captions that aim to describe everything in the image, so we'll use https://google.github.io/docci/ dataset instead
- Need to incorporate OCR to include context based off the text in the images
- Explore any models which are more 'powerful' than GPT2 when it comes to learning from images

Methodology:

Initially I just wanted a working code in which I can swap out datasets and models to see any potential improvement in performance

Converting the dataset into tokenised labels and pixel values for batch processing and faster data loading – this took a few hours (Original dataset:25.1GB -> Processed dataset:370GB)

Option to change up mean and std of feature extractors but I used what was recommended in the base code

Also had option to finetune the GPT2 parameters but I used the default settings

Initially I started off trying to train the model on entire dataset but that would have taken days so trained on 1000 images and evaluated on 1000 images (approximately 3-4 hours) and then compared performance of this partially trained model to a model trained on the entire dataset (sourced from [1]).

At first the plan was to keep increasing training set of the partially trained model until the partially trained model gives 'good enough' answers without needing to be trained on the entire dataset – but then I realised I need to check how good a fully trained model would be. That's why I started using client's model as a reference by testing both the partially trained and fully trained models on images sent by client and comparing them to the response from client's model.

Evaluation Results for the partially trained model

At Step 250 (End of Epoch 1)

- **Evaluation Loss**: 0.3013
- **Evaluation Rouge1**: 21.8406
- **Evaluation Rouge2**: 2.4336
- **Evaluation RougeL**: 20.0572
- **Evaluation RougeLsum**: 20.0791

- **Evaluation Runtime**: 1257.5095 seconds
- **Evaluation Samples per Second**: 0.795
- **Evaluation Steps per Second**: 0.199

At Step 500 (End of Epoch 2)

- **Gradient Norm**: 2.2458
- **Learning Rate**: 1.6667e-05
- **Loss**: 0.3365

Training loss of 0.3365 and evaluation loss of 0.3013 are both low, given that the usual range for cross entropy loss for image captioning is between 0.3 to 1.0 [2].

The higher ROUGE-1 (overlap of individual words) compared to ROUGE-2 (overlap of word pairs) suggests that the model is better at predicting individual words than consecutive word pairs, which is typical in earlier stages of training.

Based off these metrics alone, improvements for the partially trained model could be:

- Training for more epochs
- Being trained on a larger proportion of the overall dataset
- Tuning hyperparameters

References:

[1] huggingface.co. (n.d.). nlpconnect/vit-gpt2-image-captioning · Hugging Face. [online] Available at: https://huggingface.co/nlpconnect/vit-gpt2-image-captioning.

[2] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv preprint arXiv:1502.03044.