Image Captioning With pre-trained models

**Description of Project:**

The project leverages pre-trained image captioning models to generate captions for new images. This involves utilizing the VisionEncoderDecoderModel from Hugging Face, which combines a Vision Transformer (ViT) for image encoding and GPT-2 for text decoding.

**Brief Theory about Algorithm and Dataset Used:**

**Algorithm:**

1. **Model Initialization:**
   * Load the pre-trained VisionEncoderDecoderModel, ViTImageProcessor, and GPT2TokenizerFast from Hugging Face.
2. **Image Preprocessing:**
   * The input image is processed using the ViTImageProcessor to transform it into the required tensor format. This involves resizing the image, normalizing pixel values, and converting it into a tensor suitable for the ViT model.
3. **Caption Generation:**
   * Generate captions using the pre-trained model in either greedy or sampling mode:
     + **Greedy Mode:** Selects the highest probability token at each step, ensuring a deterministic output.
     + **Sampling Mode:** Introduces randomness by choosing from the top-k probable tokens, which can result in more diverse and creative captions.
4. **Training on Custom Dataset (Optional):**
   * If a custom dataset is used, the model can be fine-tuned to improve performance on specific types of images and captions. In this case, the A3DS dataset was used.
     + **Dataset Loading:**
       - Load and preprocess the A3DS dataset, which includes images and corresponding captions. This involves reading images and captions, and preparing them for training.
     + **Tokenization and Embedding:**
       - Tokenize the captions and convert them into tensor format for model input. This step involves mapping words to their respective indices in the vocabulary.
     + **Model Training:**
       - Fine-tune the model using the custom dataset. The training loop involves:
         1. **Retrieving Image and Caption Pairs:** Obtain image-caption pairs from the dataset for training.
         2. **Preprocessing the Images:** Convert images into the required tensor format using the ViTImageProcessor.
         3. **Tokenizing the Captions:** Convert the captions into sequences of tokens (word indices) using the tokenizer.
         4. **Calculating the Loss and Performing Backpropagation:** Compute the loss between the predicted and actual captions, and update model parameters using backpropagation.
         5. **Updating the Model Parameters Using the Optimizer:** Adjust the model weights to minimize the loss, iterating over the dataset multiple times (epochs).

**Inputs:**

* **Images to be Captioned:** New images that require caption generation.
* **Pre-trained Model Weights:** Weights for both the image encoder (ViT) and text decoder (GPT-2).
* **(Optional) Custom Dataset for Fine-Tuning:** A dataset containing images and corresponding captions for fine-tuning the model.

**Outputs:**

* **Generated Captions for the Input Images:** Textual descriptions generated by the model for the given images.

**Conditions:**

* **Format of Input Images:** The input images must be in a format supported by the ViTImageProcessor (e.g., JPG, PNG).
* **Custom Dataset Formatting:** If fine-tuning, the custom dataset should be properly formatted and preprocessed, with images and corresponding captions correctly mapped.

**Libraries:**

* transformers for VisionEncoderDecoderModel, ViTImageProcessor, and GPT2TokenizerFast.
* torch for PyTorch functionalities, including model training and tensor operations.
* numpy for numerical operations, such as array manipulations.
* PIL for image processing, allowing for loading and manipulating images.
* requests for downloading images from URLs.
* pickle for loading datasets stored in binary format.
* matplotlib for displaying images within the notebook.
* os for file operations, such as checking file existence and directory management.
* tqdm for displaying progress bars during training loops.
* warnings to suppress unnecessary warnings during execution.

**Dataset:**

* **A3DS Dataset:**
  + **Content:** Consists of images and corresponding captions, providing a structured dataset for training and evaluation.
  + **Pre-computed Features:** Includes pre-computed features using ResNet50, which can be used for efficient image processing.
  + **Vocabulary File:** Contains a vocabulary file for tokenizing captions, mapping words to indices for model input.