Assignment Two: AI-Powered Meme Generator

Objective:

create an AI that generates humorous captions for a set of provided images (memes). This assignment will help you evaluate your understanding of generative models and your ability to apply AI creatively.

Details:

Dataset:

• You can use any publicly available images for memes or create a small set of images yourself.

• Optionally, you may use an existing dataset with images and captions (e.g., meme datasets from Kaggle).

Requirements

To execute this script, the following dependencies need to be installed:

- transformers (for BLIP model)
- PIL (for image processing)
- matplotlib (for visualization)
- os (for file management)

These can be installed using the following command:

```
pip install transformers pillow matplotlib torch
```

Code Implementation

1. Importing Required Libraries

The necessary libraries are imported at the beginning:

```
from transformers import BlipProcessor, BlipForConditionalGeneration
from PIL import Image
import matplotlib.pyplot as plt
import os
```

2. Loading the BLIP Model and Processor

The BLIP model and processor are loaded from the Hugging Face model repository:

```
# Load BLIP model and processor

processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")

model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base")
```

3. Function to Generate Captions

A function is defined to generate captions based on a given image:

```
def generate_caption(image_path, prompt="A funny meme about "):

  image = Image.open(image_path).convert("RGB")

  inputs = processor(images=image, text=prompt, return_tensors="pt")

  out = model.generate(**inputs, max_length=50)

  caption = processor.decode(out[0], skip_special_tokens=True)

  return caption
```

4. Function to Display Meme with Caption

Another function is created to display the meme along with its generated caption:

```
def display_meme(image_path, caption):

    image = Image.open(image_path)

    plt.imshow(image)

    plt.axis('off')

    plt.title(caption, fontsize=16, color='red', weight='bold')

   plt.show()
```

5. Processing Meme Images

The script loops through all images in a specified directory (meme_folder), generates captions, and displays the memes:

```
# Directory containing meme images

meme_folder = "/bin/meme"
```

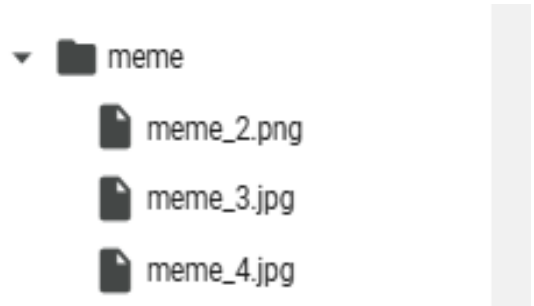
```
# Loop through images and generate memes
for image_file in os.listdir(meme_folder):
if image_file.endswith((".jpg", ".png", ".jpeg")):
        image_path = os.path.join(meme_folder, image_file)
        caption = generate_caption(image_path, prompt="A funny meme about ")
        print(f"Image: {image_file}\nCaption: {caption}\n")
        display_meme(image_path, caption)
```

Execution Steps

1. Ensure the required Python libraries are installed.
2. Store the meme images in the meme_folder directory.
3. Run the script to generate captions and display the memes with their captions.
4. Modify the prompt parameter in generate_caption() to create memes with different themes.

Output :-

1. **Store the meme images in the meme_folder directory.**



```
Image: meme_4.jpg
Caption: a funny meme about the internet
```

a funny meme about the internet


```
Image: meme_3.jpg
Caption: a funny meme about a man who is a man
```

a funny meme about a man who is a man

Image: meme_2.png
Caption: a funny meme about the fact you can ' t - a picture of a funny meme about the fact you can ' t

a funny meme about the fact you can ' t - a picture of a funny meme about the fact you can ' t