

Image Captioning Project Documentation

✓ Project Overview

This project demonstrates the use of a pre-trained image captioning model to generate descriptive captions for images. The project leverages the BLIP (Bootstrapped Language Image Pre-training) model from Salesforce, which is designed to understand and describe images in natural language.

Objectives

Utilize a pre-trained BLIP model for generating captions from images. Implement a streamlined process to handle and generate captions for individual images. Ensure the code is optimized to run on a CPU for ease of use in low-resource environments.

Tools and Libraries

Python: Programming language used for the implementation. Transformers: Hugging Face library for accessing pre-trained models. Pillow (PIL): Python Imaging Library for image processing. Torch: PyTorch library for handling tensor operations and model inference.

```
#pip install transformers torch torchvision
```

```
import torch
from transformers import BlipProcessor, BlipForConditionalGeneration
from PIL import Image
```

```
# Load pre-trained model and processor
processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base")
```

```
# Path to your image file
image_file_path = r"/content/drive/MyDrive/example.jpg"
```

```
# Function to generate captions for an image
def generate_captions(image_path):
    image = Image.open(image_path).convert("RGB")
    inputs = processor(images=image, return_tensors="pt")
    outputs = model.generate(**inputs)
    return processor.decode(outputs[0], skip_special_tokens=True)
```

```
# Generate and print caption for the single image
caption = generate_captions(image_file_path)
print(f"Image: {image_file_path}\nCaption: {caption}\n")
```

```
⚡ /usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1249: UserWarning: Using the model-agnostic default `max_length` of 20 for generation. To silence this warning, please use the `max_length` argument in the generation configuration.
warnings.warn(
Image: /content/drive/MyDrive/example.jpg
Caption: a dog is walking on the beach with a frumbnt
```

```
#the below code id for accesing the folder
```

```
import torch
from transformers import BlipProcessor, BlipForConditionalGeneration
from PIL import Image
import os

# Load pre-trained model and processor
processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base")

# Path to your small dataset
image_folder = "path_to_your_small_image_folder"
# Function to generate captions for images
def generate_captions(image_path):
    image = Image.open(image_path).convert("RGB")
    inputs = processor(images=image, return_tensors="pt")
    outputs = model.generate(**inputs)
    return processor.decode(outputs[0], skip_special_tokens=True)

# Process all images in the folder
for image_file in os.listdir(image_folder):
    if image_file.lower().endswith(('.png', '.jpg', '.jpeg')):
        image_path = os.path.join(image_folder, image_file)
        caption = generate_captions(image_path)
        print(f"Image: {image_file}\nCaption: {caption}\n")
```