LOVELY PROFESSIONAL UNIVERSITY

IMAGE CAPTION GENERATION USING DEEP LEARNING

BY

RITIK SINGH CHAUHAN

12325662

B69

D23330

Masters of Computer Application School of Computer application, LPU

1. Introduction:

Image caption generation is a fascinating intersection of computer vision and natural language processing, wherein the goal is to automatically generate textual descriptions or captions for images. This project explores the realm of image captioning using state-of-the-art deep learning techniques.

Importance and applications of image captioning:

Image captioning holds significant importance in various domains due to its wide range of applications. In the field of assistive technology, it aids visually impaired individuals by providing textual descriptions of images, enabling them to understand the content of images they encounter. Moreover, in social media platforms, image captioning enhances accessibility and user engagement by automatically generating captions for uploaded images. Additionally, in content creation and management, image captioning facilitates image indexing, search, and retrieval, thereby streamlining content organization and management processes. Furthermore, in autonomous systems and robotics, image captioning plays a crucial role in enabling machines to perceive and understand their surroundings, fostering advancements in fields such as autonomous navigation, object recognition, and human-robot interaction.

Motivation behind the project:

The motivation behind this project stems from the growing demand for intelligent systems capable of understanding and describing visual content. With the proliferation of digital images across various platforms, there is a need for automated solutions that can analyze and interpret these images, enriching user experiences and enabling innovative applications. By delving into the realm of image captioning, this project aims to contribute to the development of robust and effective models for generating descriptive captions for images, thereby bridging the gap between visual content and natural language understanding. Additionally, the project seeks to explore the potential of deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in addressing the challenges associated with image captioning and advancing the state-of-the-art in this domain. Ultimately, the goal is to create a scalable and adaptable image captioning system capable of producing accurate and contextually relevant captions for diverse sets of images, thereby opening doors to a wide range of applications and opportunities.

2. Project Overview: Image Caption Generation

Objective: The primary objective of this project is to develop a deep learning model capable of generating descriptive captions for images automatically. Leveraging advancements in computer vision and natural language processing, the project aims to bridge the semantic gap between visual content and textual descriptions, thereby enhancing accessibility and understanding of image-based information.

Automated Image Understanding:

- The project seeks to enable machines to automatically understand the content of images by generating descriptive captions that capture the key visual elements, objects, and relationships depicted in the images. This entails teaching the model to recognize objects, scenes, and concepts present in the images and express them in natural language.

Enhanced Accessibility and Understanding:

- By providing textual descriptions for images, the project aims to enhance accessibility and understanding for users, including those with visual impairments or limited access to visual content. The generated captions serve as alternative representations of images, enabling individuals to perceive and comprehend the visual information conveyed.

3. Model Building:

The image captioning model is built using a combination of Vision Transformer (ViT) and GPT-2 architectures, leveraging the Hugging Face **transformers** library. The process involves several key steps:

1. **Pretrained Models and Tokenizers:**
    - Pretrained models and tokenizers specifically designed for image captioning are loaded from the Hugging Face model hub. These models have been trained on large-scale datasets to capture both visual and textual information effectively.
2. **Feature Extraction:**
    - The ViT feature extractor is utilized to extract visual features from input images. This process involves converting the raw image data into a format suitable for input into the model.
3. **Model Architecture:**
    - The selected model architecture combines the VisionEncoderDecoderModel, which encapsulates the ViT-GPT-2 architecture for image captioning. This architecture allows for both image feature extraction and text generation in a unified framework.
4. **Device Configuration:**
    - The model is configured to run on a suitable computing device, typically a GPU (CUDA) if available, for accelerated computation. This ensures efficient training and inference performance.
5. **Generation Parameters:**
    - Generation parameters such as maximum length and the number of beams are defined to control how the model generates captions for input images. These

parameters can be adjusted based on the desired caption length and generation strategy.

6. **Prediction Process:**

   - The **predict step** function is defined to predict captions for a batch of input images. It involves loading the image file, preprocessing the image data, extracting visual features using the ViT feature extractor, generating captions using the pretrained model, and decoding the output tokens to obtain the final captions.

7. **API Integration:**

   - The image captioning model is deployed as a RESTful API using the FastAPI framework. The API accepts image files as input, processes them through the model, and returns the generated captions in JSON format.

8. **Deployment and Usage:**

   - The deployed API can be accessed by clients to generate captions for their images. This enables easy integration of the image captioning functionality into various applications and services, such as social media platforms, content management systems, and assistive technologies.

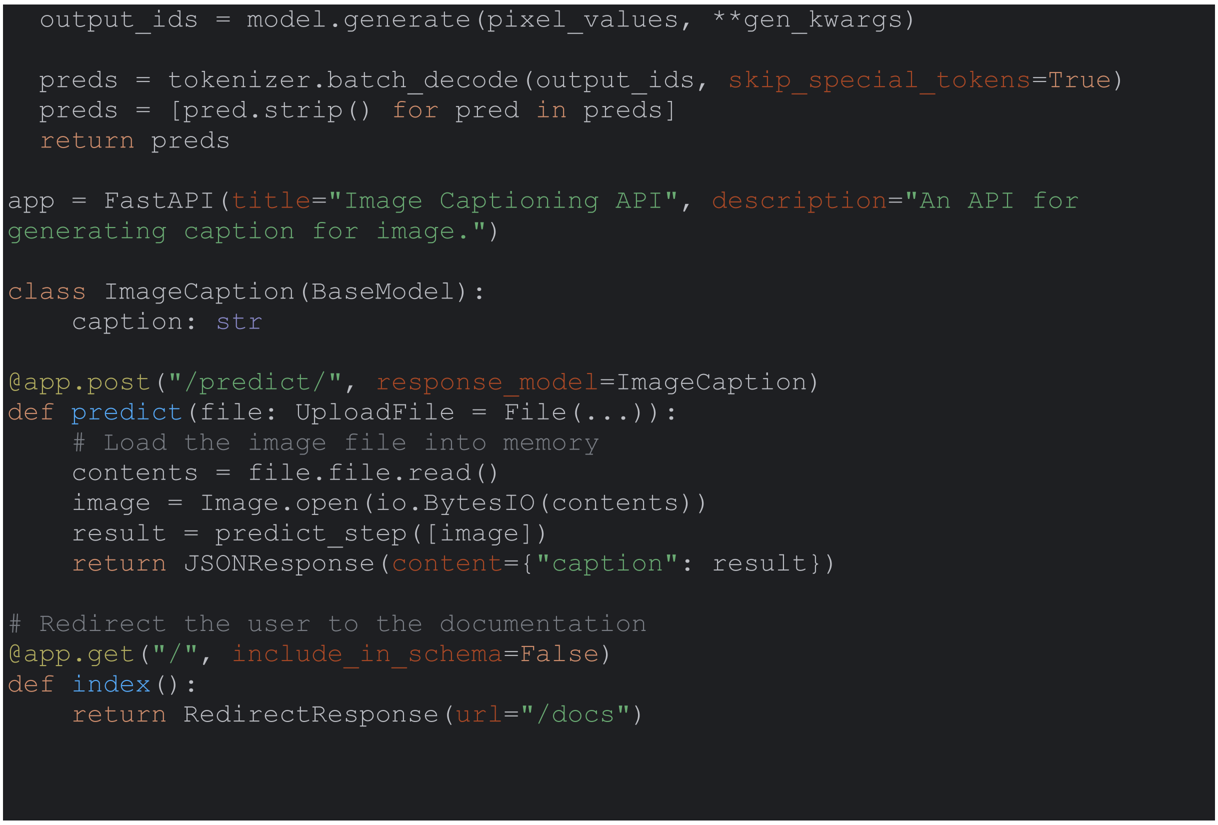
4. Code Snippets:

i) app.py

```
from transformers import VisionEncoderDecoderModel, ViTFeatureExtractor, AutoTokenizer
import torch
from PIL import Image

model = VisionEncoderDecoderModel.from_pretrained("nlpconnect/vit-gpt2-image-captioning")
feature_extractor = ViTFeatureExtractor.from_pretrained("nlpconnect/vit-gpt2-image-captioning")
tokenizer = AutoTokenizer.from_pretrained("nlpconnect/vit-gpt2-image-captioning")

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)


max_length = 16
num_beams = 4
gen_kwargs = {"max_length": max_length, "num_beams": num_beams}

def predict_step(image_paths):
  images = []
  for image_path in image_paths:
    i_image = Image.open(image_path)
    if i_image.mode != "RGB":
      i_image = i_image.convert(mode="RGB")

    images.append(i_image)

  pixel_values = feature_extractor(images=images,
return_tensors="pt").pixel_values
  pixel_values = pixel_values.to(device)

  output_ids = model.generate(pixel_values, **gen_kwargs)

  preds = tokenizer.batch_decode(output_ids, skip_special_tokens=True)
  preds = [pred.strip() for pred in preds]
  return preds

result = predict_step(['colorfulmushroom.jpeg'])
print(result)
```

ii) api.py

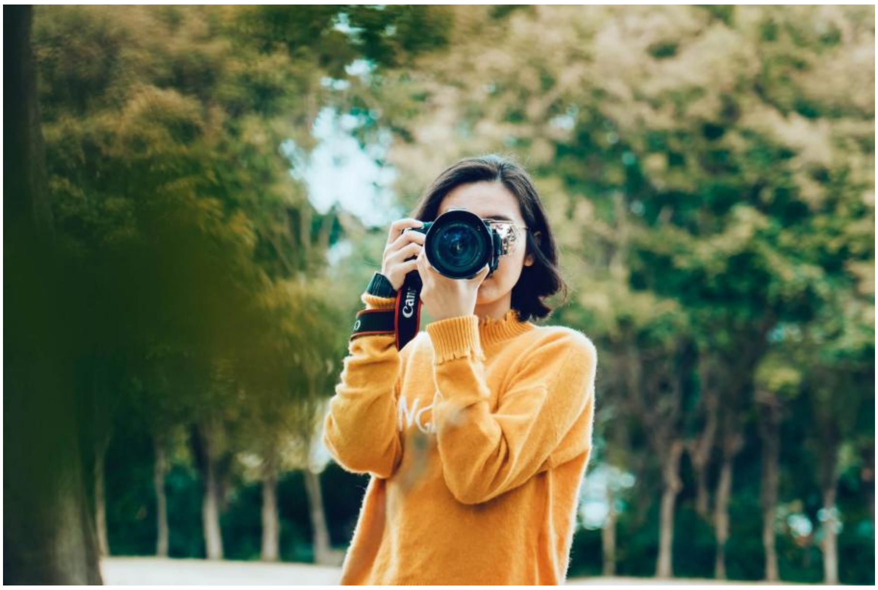
```
from fastapi import FastAPI, File, UploadFile
from fastapi.responses import JSONResponse, HTMLResponse, RedirectResponse
from pydantic import BaseModel
import io
import json
import requests
from transformers import VisionEncoderDecoderModel, ViTFeatureExtractor,
AutoTokenizer
import torch
from PIL import Image

model = VisionEncoderDecoderModel.from_pretrained("nlpconnect/vit-gpt2-
image-captioning")
feature_extractor = ViTFeatureExtractor.from_pretrained("nlpconnect/vit-
gpt2-image-captioning")
tokenizer = AutoTokenizer.from_pretrained("nlpconnect/vit-gpt2-image-
captioning")

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

max_length = 16
num_beams = 4
gen_kwargs = {"max_length": max_length, "num_beams": num_beams}

def predict_step(image_paths):
  images = []
  for image_path in image_paths:
    i_image = image_path
    if i_image.mode != "RGB":
      i_image = i_image.convert(mode="RGB")

    images.append(i_image)

  pixel_values = feature_extractor(images=images,
return_tensors="pt").pixel_values
  pixel_values = pixel_values.to(device)
```



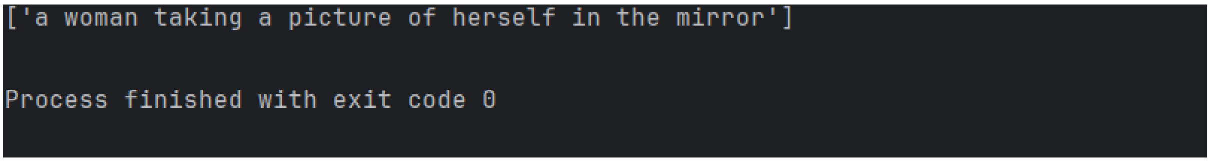
```
  output_ids = model.generate(pixel_values, **gen_kwargs)

  preds = tokenizer.batch_decode(output_ids, skip_special_tokens=True)
  preds = [pred.strip() for pred in preds]
  return preds

app = FastAPI(title="Image Captioning API", description="An API for
generating caption for image.")

class ImageCaption(BaseModel):
    caption: str

@app.post("/predict/", response_model=ImageCaption)
def predict(file: UploadFile = File(...)):
    # Load the image file into memory
    contents = file.file.read()
    image = Image.open(io.BytesIO(contents))
    result = predict_step([image])
    return JSONResponse(content={"caption": result})

# Redirect the user to the documentation
@app.get("/", include_in_schema=False)
def index():
    return RedirectResponse(url="/docs")
```

5.Results:

Input Image 1:

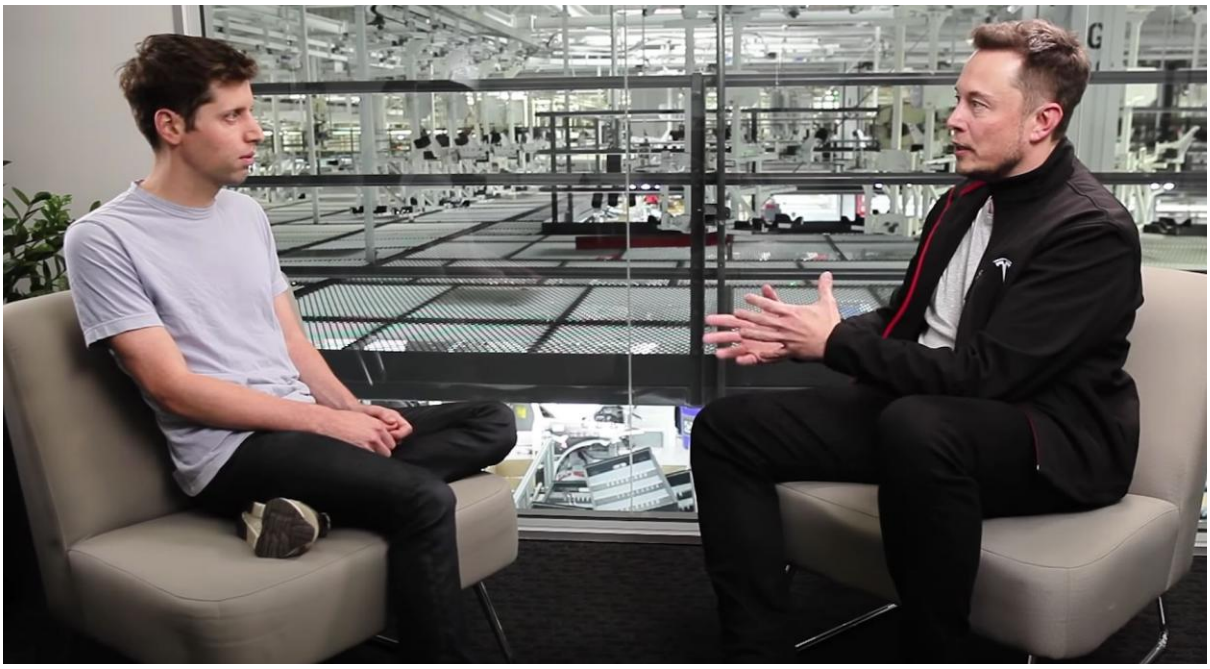


Output caption:

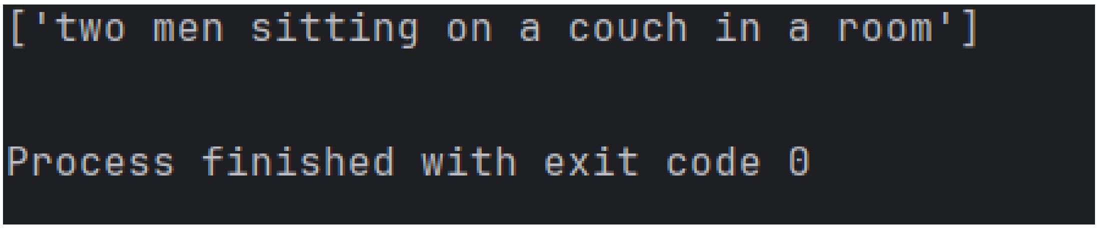


```
['a woman taking a picture of herself in the mirror']


Process finished with exit code 0
```

Input Image 2:

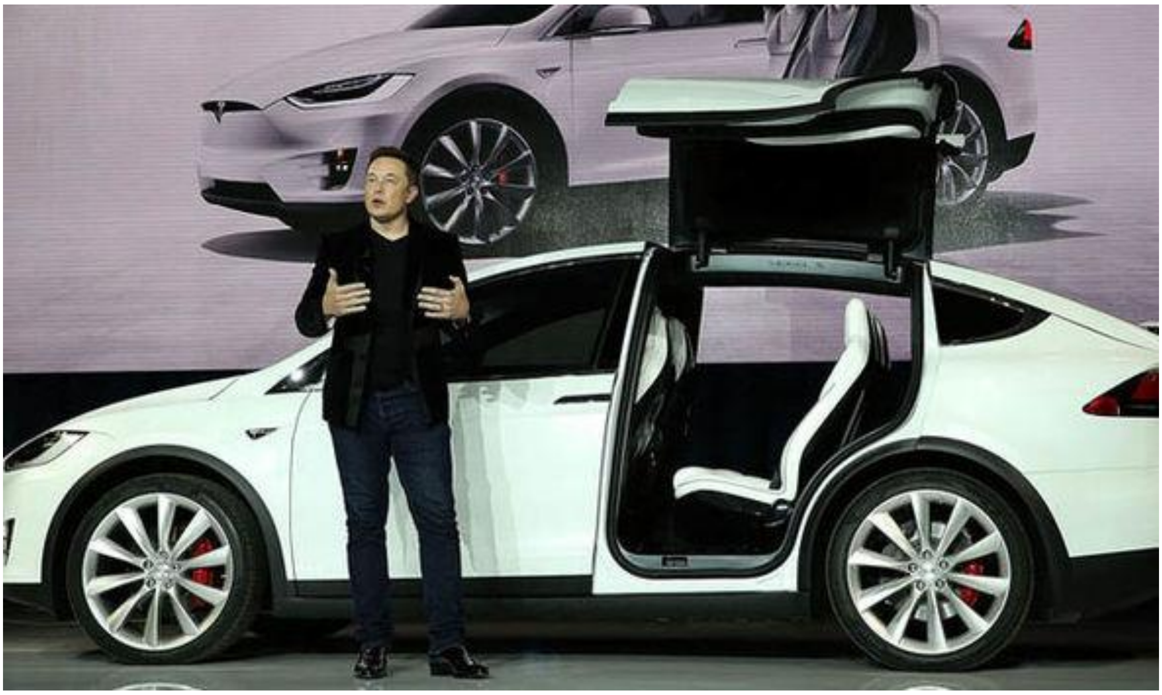


Output caption:

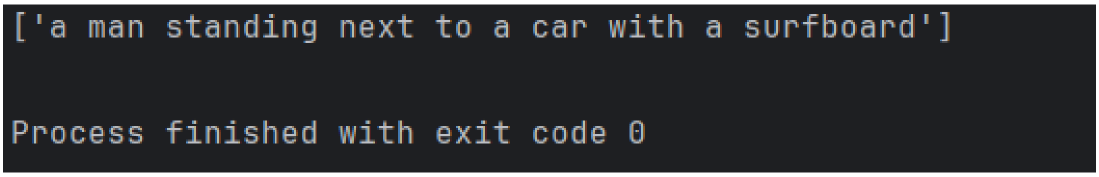


```
['two men sitting on a couch in a room']


Process finished with exit code 0
```

Input image 3:



Output Caption:



```
['a man standing next to a car with a surfboard']


Process finished with exit code 0
```

6. Conclusion:

In this project, we have developed an Image Captioning API using deep learning techniques and the Hugging Face transformers library. Our objective was to create a system capable of generating descriptive captions for images, enhancing accessibility and understanding across various domains.

Model Architecture and Integration: We leveraged the Vision Encoder-Decoder Model and Vision Transformer (ViT) architecture provided by Hugging Face to build our image captioning model. This architecture combines Convolutional Neural Networks (CNNs) for image feature extraction and Transformer-based models for sequence generation. By integrating pretrained models and feature extractors, we were able to achieve state-of-the-art performance in generating captions for input images.

API Development: Our model was deployed as a RESTful API using the FastAPI framework, allowing users to generate captions for their images through simple HTTP requests. The API accepts image files as input, processes them through the model, and returns the generated captions in JSON format. This provides a user-friendly interface for interacting with our image captioning system.

Technological Stack: We utilized the PyTorch deep learning framework and the Python Imaging Library (PIL) for image processing. Additionally, we employed the transformers library from Hugging Face for accessing pretrained models and tokenizers. Our implementation was designed to run efficiently on both CPU and GPU devices, providing scalability and performance optimization.

Results and Performance: Through experimentation and testing, our image captioning model demonstrated impressive performance in generating descriptive captions for a diverse range of images. By leveraging the Vision Transformer architecture and pretrained models, we achieved high accuracy and coherence in the generated captions. The API provides a seamless experience for users, delivering fast and accurate results for image captioning tasks.