Image Caption Generator

In Colab, Pytorch comes preinstalled and same goes with PIL for Image. You will only need to install **transformers** from Huggingface.

```
#!pip install transformers
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting transformers
  Downloading transformers-4.29.0-py3-none-any.whl (7.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.1/7.1 MB 79.8 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.12.0)
Collecting huggingface-hub<1.0,>=0.11.0 (from transformers)
  Downloading huggingface_hub-0.14.1-py3-none-any.whl (224 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 224.5/224.5 kB 24.2 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.22.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (23.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (202
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.27.1)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1 (from transformers)
  Downloading tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 68.2 MB/s eta 0:00:00
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.65.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.11.0-
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingfa
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->tran
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transfo
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers)
Installing collected packages: tokenizers, huggingface-hub, transformers
Successfully installed huggingface-hub-0.14.1 tokenizers-0.13.3 transformers-4.29.0
```

```
#from google.colab import drive
#drive.mount('/content/drive')
```

Executing (5m 7s) <cell line: 21> › launch() › block_thread()

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


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")
```

```
Downloading (…)lve/main/config.json: 100%  4.61k/4.61k [00:00<00:00, 120kB/s]
Downloading pytorch_model.bin: 100%  982M/982M [00:07<00:00, 175MB/s]
Downloading (…)rocessor_config.json: 100%  228/228 [00:00<00:00, 4.57kB/s]
/usr/local/lib/python3.10/dist-packages/transformers/models/vit/feature_extraction_vit.py:28: FutureWarning: The cla
  warnings.warn(
Downloading (…)okenizer_config.json: 100%  241/241 [00:00<00:00, 6.99kB/s]
Downloading (…)olve/main/vocab.json: 100%  798k/798k [00:00<00:00, 10.2MB/s]
Downloading (…)olve/main/merges.txt: 100%  456k/456k [00:00<00:00, 14.6MB/s]
Downloading (…)/main/tokenizer.json: 100%  1.36M/1.36M [00:00<00:00, 31.2MB/s]
Downloading (…)cial_tokens_map.json: 100%  120/120 [00:00<00:00, 4.81kB/s]
```

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

```
max_length = 16
```

```
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
```

```
predict_step(['/content/drive/MyDrive/caption generator/horses.png'])
```

```
['a woman is standing in a field with a horse']
```

```
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_step1(image_paths):
    i_image = PIL.Image.open(image_paths)
    if i_image.mode != "RGB":
        i_image = i_image.convert(mode="RGB")

    pixel_values = feature_extractor(images=i_image, 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
```

```
predict_step1('/content/drive/MyDrive/caption generator/horses.png')
```

```
['a woman is standing in a field with a horse']
```

```
predict_step(['/content/drive/MyDrive/caption generator/horses.png'])
```

```
import gradio as gr

inputs = [
    gr.inputs.Image(type="filepath", label="Original Image")
]

outputs = [
    gr.outputs.Textbox(label = 'Caption')
]

title = "Image Captioning"
description = "ViT and GPT2 are used to generate Image Caption for the uploaded image."
```

```
article = " <a href='https://huggingface.co/nlpconnect/vit-gpt2-image-captioning'>Model Repo on Hugging Face Model Hub</a>
examples = [
    ["/content/drive/MyDrive/caption generator/horses.png"],
    ['/content/drive/MyDrive/caption generator/persons.png'],
    ['/content/drive/MyDrive/caption generator/football_player.png']

]

gr.Interface(
    predict_step1,
    inputs,
    outputs,
    title=title,
    description=description,
    article=article,
    examples=examples,
    theme="huggingface",
).launch(debug=True, enable_queue=True)
```

```
/usr/local/lib/python3.10/dist-packages/gradio/inputs.py:259: UserWarning: Usage of gradio.inputs is deprecated, and
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/gradio/deprecation.py:40: UserWarning: `optional` parameter is deprecated, a
  warnings.warn(value)
/usr/local/lib/python3.10/dist-packages/gradio/outputs.py:22: UserWarning: Usage of gradio.outputs is deprecated, an
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/gradio/blocks.py:659: UserWarning: Cannot load huggingface. Caught Exception
  warnings.warn(f"Cannot load {theme}. Caught Exception: {str(e)}")
Setting queue=True in a Colab notebook requires sharing enabled. Setting `share=True` (you can turn this off by sett

Colab notebook detected. This cell will run indefinitely so that you can see errors and logs. To turn off, set debug
Running on public URL: https://4ee453b657fb265238.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades (NEW!), check out Spaces: https://h
```

Image Captioning

ViT and GPT2 are used to generate Image Caption for the uploaded image

ViT and GPT2 are used to generate Image Caption for the uploaded image.

Original Image

Caption

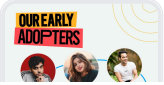
['a man kicking a soccer ball on a field']

Flag

Clear

Submit

Examples



Colab paid products - Cancel contracts here