Image Caption Generation using huggingface

In [9]:
```
pip install transformers
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheel
s/public/simple/
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-pack
ages (4.29.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages
(from transformers) (3.12.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in /usr/local/lib/pyth
on3.10/dist-packages (from transformers) (0.14.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packa
ges (from transformers) (1.22.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-p
ackages (from transformers) (23.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packa
ges (from transformers) (6.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist
-packages (from transformers) (2022.10.31)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages
(from transformers) (2.27.1)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/li
b/python3.10/dist-packages (from transformers) (0.13.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packag
es (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->transformers) (2023.4.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python
3.10/dist-packages (from huggingface-hub<1.0,>=0.11.0->transformers) (4.5.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/
dist-packages (from requests->transformers) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dis
t-packages (from requests->transformers) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python
3.10/dist-packages (from requests->transformers) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-pack
ages (from requests->transformers) (3.4)
```

In [10]:
```
from transformers import VisionEncoderDecoderModel,ViTFeatureExtractor,ViTImagePro
import cv2
from PIL import Image
import torch
```

In [11]:
```
model=VisionEncoderDecoderModel.from_pretrained("nlpconnect/vit-gpt2-image-caption
feature_extractor=ViTImageProcessor.from_pretrained("nlpconnect/vit-gpt2-image-cap
tokenizer=AutoTokenizer.from_pretrained("nlpconnect/vit-gpt2-image-captioning")
```

In [12]:
```
device=torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
```

Out[12]:

```
VisionEncoderDecoderModel(
  (encoder): ViTModel(
    (embeddings): ViTEmbeddings(
      (patch_embeddings): ViTPatchEmbeddings(
        (projection): Conv2d(3, 768, kernel_size=(16, 16), stride=(16, 16))
      )
      (dropout): Dropout(p=0.0, inplace=False)
    )
    (encoder): ViTEncoder(
      (layer): ModuleList(
        (0-11): 12 x ViTLayer(
          (attention): ViTAttention(
            (attention): ViTSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
            (output): ViTSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
          )
          (intermediate): ViTIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ViTOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (dropout): Dropout(p=0.0, inplace=False)
          )
          (layernorm_before): LayerNorm((768,), eps=1e-12, elementwise_affine=Tru
e)
          (layernorm_after): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
    (layernorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (pooler): ViTPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (decoder): GPT2LMHeadModel(
    (transformer): GPT2Model(
      (wte): Embedding(50257, 768)
      (wpe): Embedding(1024, 768)
      (drop): Dropout(p=0.1, inplace=False)
      (h): ModuleList(
        (0-11): 12 x GPT2Block(
          (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
          (attn): GPT2Attention(
            (c_attn): Conv1D()
            (c_proj): Conv1D()
            (attn_dropout): Dropout(p=0.1, inplace=False)
            (resid_dropout): Dropout(p=0.1, inplace=False)
          )
          (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
          (crossattention): GPT2Attention(
            (c_attn): Conv1D()
            (q_attn): Conv1D()
            (c_proj): Conv1D()
            (attn_dropout): Dropout(p=0.1, inplace=False)
            (resid_dropout): Dropout(p=0.1, inplace=False)
```

```
            )
            (ln_cross_attn): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
            (mlp): GPT2MLP(
              (c_fc): Conv1D()
              (c_proj): Conv1D()
              (act): NewGELUActivation()
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
        )
        (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
      )
      (lm_head): Linear(in_features=768, out_features=50257, bias=False)
    )
)
```

In [13]:
```
def image_predict(img):
  pixel_values=feature_extractor(images=[img],return_tensors="pt").pixel_values
  pixel_values=pixel_values.to(device)
  max_length=128
  output=model.generate(pixel_values,num_beams=4,max_length=max_length)
  preds=tokenizer.decode(output[0],skip_special_tokens=True)
  print(preds)
```

In [15]:
```
from google.colab.patches import cv2_imshow
image_path="/content/Image1.png"
image=cv2.imread(image_path)
image=cv2.resize(image,(500,500))
cv2_imshow(image)
image_predict(image)
```

```
a man kicking a soccer ball on a field
```

In [16]:
```
image_path="/content/Image2.png"
image=cv2.imread(image_path)
image=cv2.resize(image,(500,500))
cv2_imshow(image)
image_predict(image)
```

```
a woman standing on top of a horse in a field
```

In [19]:
```
image_path="/content/Image3.png"
image=cv2.imread(image_path)
image=cv2.resize(image,(500,500))
cv2_imshow(image)
image_predict(image)
```

a collage of photos showing different types of food