Automode LoRA

Abstract

This method generates descriptions for target images, enabling an end-to-end annotation workflow that significantly reduces manual effort. By combining this automated annotation with the LoRA technique, our approach not only eliminates the need for manual annotation but also minimizes the number of parameters that require training.

Run Image Captioning

To get dataset for LoRA

Download Model

```
import requests
from PIL import Image
from transformers import BlipProcessor, BlipForConditionalGeneration
import warnings

warnings.filterwarnings("ignore")
processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-large")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-large")
```

Load Your Image

```
img_url = 'https://rotomlabs.net/_next/image?url=https%3A%2F%2Fstatic.rotomlabs.net%2Fimages%2Fofficial-artwork%2F0630-mandibuzz.webp&w=750&q=75'
raw_image = Image.open("../pokemonDataset/pokemon/a bird with a large beak and a large beak.jpg").convert('RGB')
raw_image

```

Get Description

```
# conditional image captioning
text = "a photography of"
inputs = processor(raw_image, text, return_tensors="pt")
out = model.generate(**inputs)
print(processor.decode(out[0], skip_special_tokens=True))
```

output: a photography of a very large bird with a very long beak

Run LoRA

```
!python "main.py" --project_name "Dreambooth_dog" --training_model "C:\\Users\\Administrator\\Downloads\\sd-v1-4.ckpt" --regularization_images "C:\\Users\\Administrator\\Desktop\\Dreambooth-Stable-Diffusion-main\\image\\regularization\\regularization_rabbit" --training_images "C:\\Users\\Administrator\\Desktop\\Dreambooth-Stable-Diffusion-main\\image\\samples\\rabbit" --max_training_steps 2000 --class_word "rabbit" --token "tiantian" --flip_p 0 --learning_rate 1.0e-06 --save_every_x_steps 500
```