import os

from PIL import Image, ImageDraw, ImageFont

from transformers import BlipProcessor, BlipForConditionalGeneration

# Define the directory containing the images

image\_dir = r'<Path of images folder>'

# Load BLIP model and processor

model\_name = "Salesforce/blip-image-captioning-base"

processor = BlipProcessor.from\_pretrained(model\_name)

model = BlipForConditionalGeneration.from\_pretrained(model\_name)

# Function to generate description using BLIP

def generate\_description(image\_path):

with open(image\_path, "rb") as image\_file:

image = Image.open(image\_file).convert("RGB")

inputs = processor(images=image, return\_tensors="pt")

outputs = model.generate(\*\*inputs)

description = processor.decode(outputs[0], skip\_special\_tokens=True)

return description

# Function to add description to an image

def add\_description\_to\_image(image\_path, description, output\_path):

# Open an image file

with Image.open(image\_path) as img:

# Initialize ImageDraw

draw = ImageDraw.Draw(img)

# Use a truetype font

font = ImageFont.load\_default()

# Position for the description

text\_position = (10, 10)

# Add text to image

draw.text(text\_position, description, font=font, fill="white")

# Save the edited image

img.save(output\_path)

# Iterate over all images in the directory

for filename in os.listdir(image\_dir):

if filename.endswith(".jpg") or filename.endswith(".png"):

image\_path = os.path.join(image\_dir, filename)

output\_path = os.path.join(image\_dir, "annotated\_" + filename)

description = generate\_description(image\_path)

add\_description\_to\_image(image\_path, description, output\_path)