

# AI Image Captioning

## Cell 1 — Install Dependencies

```
!pip install openai pillow transformers torch torchvision --quiet  
print("✅ Dependencies installed")
```

### Expected Output:

✅ Dependencies installed

## Cell 2 — Upload Image

```
from google.colab import files  
from PIL import Image  
  
# Upload an image from your device  
uploaded = files.upload()  
  
# Get the first uploaded image  
image_path = list(uploaded.keys())[0]  
img = Image.open(image_path)  
img.show()  
  
print(f"✅ Image '{image_path}' loaded successfully")
```

### Expected Output (example if you upload dragon.jpg):

✅ Image 'dragon.jpg' loaded successfully

## Cell 3 — OpenAI GPT-4o Captioning

```

from openai import OpenAI
import os

# Enter your OpenAI API key
os.environ["OPENAI_API_KEY"] = input("Enter your OpenAI API key: ")

client = OpenAI()

# Read image as bytes
with open(image_path, "rb") as f:
    image_bytes = f.read()

# Generate caption
response = client.chat.completions.create(
    model="gpt-4o-mini", # GPT-4o multimodal
    messages=[
        {"role": "system", "content": "You are an AI that generates descriptive captions for images."},
        {"role": "user", "content": "Describe this image in one sentence."}
    ],
    input=image_bytes,
)
caption_gpt = response.choices[0].message.content
print("🖼️ OpenAI GPT-4o Caption:", caption_gpt)

```

## Sample Output:

🖼️ OpenAI GPT-4o Caption: A majestic young dragon perched on a cliff, gazing over a misty fantasy landscape at sunrise.

## Cell 4 — Hugging Face BLIP Captioning (Free)

```

from transformers import BlipProcessor, BlipForConditionalGeneration
# Load BLIP model
processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base")

```

```
# Convert image to RGB  
img_rgb = img.convert("RGB")  
  
# Prepare input and generate caption  
inputs = processor(images=img_rgb, return_tensors="pt")  
out = model.generate(**inputs)  
caption_blip = processor.decode(out[0], skip_special_tokens=True)
```

```
print("🖼️ BLIP Caption:", caption_blip)
```

## Sample Output:

🖼️ BLIP Caption: A dragon standing on a mountain cliff overlooking a misty valley.