

```
!pip install -q transformers pillow torch
```

```
import torch
from transformers import BlipProcessor, BlipForConditionalGeneration
from PIL import Image

# Load the model (this may take a minute the first time)
processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base").to("cuda")

def generate_prompt(image_path):
    raw_image = Image.open(image_path).convert('RGB')

    # Process image for the model
    inputs = processor(raw_image, return_tensors="pt").to("cuda")

    # Generate caption
    out = model.generate(**inputs)
    caption = processor.decode(out[0], skip_special_tokens=True)

    # Add your "Prompt Optimization" keywords
    optimized_prompt = f"{caption}, highly detailed, cinematic lighting, 8k, masterpiece style"
    negative_prompt = "blurry, distorted, low quality, watermark, extra fingers"

    return optimized_prompt, negative_prompt
```

Using a slow image processor as `use_fast` is unset and a slow processor was saved with this model. `use_fast=True` will be the default in the future.
 /usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
 The secret `HF_TOKEN` does not exist in your Colab secrets.
 To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as an environment variable `HF_TOKEN` or add it to your Colab secrets.
 You will be able to reuse this secret in all of your notebooks.
 Please note that authentication is recommended but still optional to access public models or datasets.

```
warnings.warn(
preprocessor_config.json: 100% 287/287 [00:00<00:00, 26.6kB/s]
tokenizer_config.json: 100% 506/506 [00:00<00:00, 55.1kB/s]
vocab.txt: 232k/? [00:00<00:00, 18.0MB/s]
tokenizer.json: 711k/? [00:00<00:00, 37.9MB/s]
special_tokens_map.json: 100% 125/125 [00:00<00:00, 14.2kB/s]
config.json: 4.56k/? [00:00<00:00, 340kB/s]
pytorch_model.bin: 100% 990M/990M [00:07<00:00, 286MB/s]
```

```
# Replace 'WhatsApp Image 2025-12-24 at 8.09.30 AM' with the name of the file you uploaded
opt_p, neg_p = generate_prompt('WhatsApp Image 2025-12-24 at 8.09.30 AM')
```

```
print(f"🔥 Optimized Prompt: {opt_p}")
print(f"🚫 Negative Prompt: {neg_p}")
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
/tmp/ipython-input-2713777489.py in <cell line: 0>()
      1 # Replace 'WhatsApp Image 2025-12-24 at 8.09.30 AM' with the name of the file you uploaded
----> 2 opt_p, neg_p = generate_prompt('WhatsApp Image 2025-12-24 at 8.09.30 AM')
      3
      4 print(f"🔥 Optimized Prompt: {opt_p}")
      5 print(f"🚫 Negative Prompt: {neg_p}")
```

```
----- 1 frames -----
/usr/local/lib/python3.12/dist-packages/PIL/Image.py in open(fp, mode, formats)
    3511     if is_path(fp):
    3512         filename = os.fspath(fp)
-> 3513         fp = builtins.open(filename, "rb")
    3514         exclusive_fp = True
    3515     else:
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'WhatsApp Image 2025-12-24 at 8.09.30 AM'
```

```
print(f"🔥 Optimized Prompt: {opt_p}")
print(f"🚫 Negative Prompt: {neg_p}")
```

```
def generate_prompt(image_path, style="Realistic"):
    raw_image = Image.open(image_path).convert('RGB')
    inputs = processor(raw_image, return_tensors="pt").to("cuda")
    out = model.generate(**inputs)
    caption = processor.decode(out[0], skip_special_tokens=True)

    # Style Dictionary
    styles = {
        "Realistic": "highly detailed, 8k resolution, cinematic lighting, masterpiece, photorealistic",
        "Anime": "anime style, studio ghibli, vibrant colors, clean lines, high quality digital art",
        "Cyberpunk": "neon lights, futuristic, synthwave aesthetic, dark city background, high contrast"
    }

    selected_style = styles.get(style, styles["Realistic"])

    optimized_prompt = f"{caption}, {selected_style}"
    negative_prompt = "blurry, low quality, distorted, watermark, text, grainy, extra limbs"

    return optimized_prompt, negative_prompt
```

🔥 Optimized Prompt: a cat laying in the grass, anime style, studio ghibli, vibrant colors, clean lines, high quality digital

```

  3.3.1...
added 22 packages in 2s
  3.3.2...
  3.3.3...
  3.3.4...
  3.3.5...
  3.3.6...
  3.3.7...
  3.3.8...
  3.3.9...
  3.3.10...
  3.3.11...
  3.3.12...
  3.3.13...
  3.3.14...
  3.3.15...
  3.3.16...
  3.3.17...
  3.3.18...
  3.3.19...
  3.3.20...
  3.3.21...
  3.3.22...
  3.3.23...
  3.3.24...
  3.3.25...
  3.3.26...
  3.3.27...
  3.3.28...
  3.3.29...
  3.3.30...
  3.3.31...
  3.3.32...
  3.3.33...
  3.3.34...
  3.3.35...
  3.3.36...
  3.3.37...
  3.3.38...
  3.3.39...
  3.3.40...
  3.3.41...
  3.3.42...
  3.3.43...
  3.3.44...
  3.3.45...
  3.3.46...
  3.3.47...
  3.3.48...
  3.3.49...
  3.3.50...
  3.3.51...
  3.3.52...
  3.3.53...
  3.3.54...
  3.3.55...
  3.3.56...
  3.3.57...
  3.3.58...
  3.3.59...
  3.3.60...
  3.3.61...
  3.3.62...
  3.3.63...
  3.3.64...
  3.3.65...
  3.3.66...
  3.3.67...
  3.3.68...
  3.3.69...
  3.3.70...
  3.3.71...
  3.3.72...
  3.3.73...
  3.3.74...
  3.3.75...
  3.3.76...
  3.3.77...
  3.3.78...
  3.3.79...
  3.3.80...
  3.3.81...
  3.3.82...
  3.3.83...
  3.3.84...
  3.3.85...
  3.3.86...
  3.3.87...
  3.3.88...
  3.3.89...
  3.3.90...
  3.3.91...
  3.3.92...
  3.3.93...
  3.3.94...
  3.3.95...
  3.3.96...
  3.3.97...
  3.3.98...
  3.3.99...
  3.4.0...
  3.4.1...
  3.4.2...
  3.4.3...
  3.4.4...
  3.4.5...
  3.4.6...
  3.4.7...
  3.4.8...
  3.4.9...
  3.4.10...
  3.4.11...
  3.4.12...
  3.4.13...
  3.4.14...
  3.4.15...
  3.4.16...
  3.4.17...
  3.4.18...
  3.4.19...
  3.4.20...
  3.4.21...
  3.4.22...
  3.4.23...
  3.4.24...
  3.4.25...
  3.4.26...
  3.4.27...
  3.4.28...
  3.4.29...
  3.4.30...
  3.4.31...
  3.4.32...
  3.4.33...
  3.4.34...
  3.4.35...
  3.4.36...
  3.4.37...
  3.4.38...
  3.4.39...
  3.4.40...
  3.4.41...
  3.4.42...
  3.4.43...
  3.4.44...
  3.4.45...
  3.4.46...
  3.4.47...
  3.4.48...
  3.4.49...
  3.4.50...
  3.4.51...
  3.4.52...
  3.4.53...
  3.4.54...
  3.4.55...
  3.4.56...
  3.4.57...
  3.4.58...
  3.4.59...
  3.4.60...
  3.4.61...
  3.4.62...
  3.4.63...
  3.4.64...
  3.4.65...
  3.4.66...
  3.4.67...
  3.4.68...
  3.4.69...
  3.4.70...
  3.4.71...
  3.4.72...
  3.4.73...
  3.4.74...
  3.4.75...
  3.4.76...
  3.4.77...
  3.4.78...
  3.4.79...
  3.4.80...
  3.4.81...
  3.4.82...
  3.4.83...
  3.4.84...
  3.4.85...
  3.4.86...
  3.4.87...
  3.4.88...
  3.4.89...
  3.4.90...
  3.4.91...
  3.4.92...
  3.4.93...
  3.4.94...
  3.4.95...
  3.4.96...
  3.4.97...
  3.4.98...
  3.4.99...
  3.5.0...
  3.5.1...
  3.5.2...
  3.5.3...
  3.5.4...
  3.5.5...
  3.5.6...
  3.5.7...
  3.5.8...
  3.5.9...
  3.5.10...
  3.5.11...
  3.5.12...
  3.5.13...
  3.5.14...
  3.5.15...
  3.5.16...
  3.5.17...
  3.5.18...
  3.5.19...
  3.5.20...
  3.5.21...
  3.5.22...
  3.5.23...
  3.5.24...
  3.5.25...
  3.5.26...
  3.5.27...
  3.5.28...
  3.5.29...
  3.5.30...
  3.5.31...
  3.5.32...
  3.5.33...
  3.5.34...
  3.5.35...
  3.5.36...
  3.5.37...
  3.5.38...
  3.5.39...
  3.5.40...
  3.5.41...
  3.5.42...
  3.5.43...
  3.5.44...
  3.5.45...
  3.5.46...
  3.5.47...
  3.5.48...
  3.5.49...
  3.5.50...
  3.5.51...
  3.5.52...
  3.5.53...
  3.5.54...
  3.5.55...
  3.5.56...
  3.5.57...
  3.5.58...
  3.5.59...
  3.5.60...
  3.5.61...
  3.5.62...
  3.5.63...
  3.5.64...
  3.5.65...
  3.5.66...
  3.5.67...
  3.5.68...
  3.5.69...
  3.5.70...
  3.5.71...
  3.5.72...
  3.5.73...
  3.5.74...
  3.5.75...
  3.5.76...
  3.5.77...
  3.5.78...
  3.5.79...
  3.5.80...
  3.5.81...
  3.5.82...
  3.5.83...
  3.5.84...
  3.5.85...
  3.5.86...
  3.5.87...
  3.5.88...
  3.5.89...
  3.5.90...
  3.5.91...
  3.5.92...
  3.5.93...
  3.5.94...
  3.5.95...
  3.5.96...
  3.5.97...
  3.5.98...
  3.5.99...
  3.6.0...
  3.6.1...
  3.6.2...
  3.6.3...
  3.6.4...
  3.6.5...
  3.6.6...
  3.6.7...
  3.6.8...
  3.6.9...
  3.6.10...
  3.6.11...
  3.6.12...
  3.6.13...
  3.6.14...
  3.6.15...
  3.6.16...
  3.6.17...
  3.6.18...
  3.6.19...
  3.6.20...
  3.6.21...
  3.6.22...
  3.6.23...
  3.6.24...
  3.6.25...
  3.6.26...
  3.6.27...
  3.6.28...
  3.6.29...
  3.6.30...
  3.6.31...
  3.6.32...
  3.6.33...
  3.6.34...
  3.6.35...
  3.6.36...
  3.6.37...
  3.6.38...
  3.6.39...
  3.6.40...
  3.6.41...
  3.6.42...
  3.6.43...
  3.6.44...
  3.6.45...
  3.6.46...
  3.6.47...
  3.6.48...
  3.6.49...
  3.6.50...
  3.6.51...
  3.6.52...
  3.6.53...
  3.6.54...
  3.6.55...
  3.6.56...
  3.6.57...
  3.6.58...
  3.6.59...
  3.6.60...
  3.6.61...
  3.6.62...
  3.6.63...
  3.6.64...
  3.6.65...
  3.6.66...
  3.6.67...
  3.6.68...
  3.6.69...
  3.6.70...
  3.6.71...
  3.6.72...
  3.6.73...
  3.6.74...
  3.6.75...
  3.6.76...
  3.6.77...
  3.6.78...
  3.6.79...
  3.6.80...
  3.6.81...
  3.6.82...
  3.6.83...
  3.6.84...
  3.6.85...
  3.6.86...
  3.6.87...
  3.6.88...
  3.6.89...
  3.6.90...
  3.6.91...
  3.6.92...
  3.6.93...
  3.6.94...
  3.6.95...
  3.6.96...
  3.6.97...
  3.6.98...
  3.6.99...
  3.7.0...
  3.7.1...
  3.7.2...
  3.7.3...
  3.7.4...
  3.7.5...
  3.7.6...
  3.7.7...
  3.7.8...
  3.7.9...
  3.7.10...
  3.7.11...
  3.7.12...
  3.7.13...
  
```

Password/Endpoint IP for localtunnel is: 34.125.101.94

```
import subprocess
import sys

# This forces the installation if something is missing
try:
    import transformers
    import torch
except ImportError:
    subprocess.check_call([sys.executable, "-m", "pip", "install", "transformers", "torch", "Pillow"])
```

```
%%writefile app.py
import streamlit as st
import torch
from transformers import BlipProcessor, BlipForConditionalGeneration
from PIL import Image

# 1. Page Configuration for a professional look
st.set_page_config(page_title="Vision-Prompt AI", page_icon="🧠", layout="wide")

# 2. Custom CSS for colors and styling
st.markdown("""
<style>
```

```

        background-color: #f0f2f6;
    }
    .stButton>button {
        width: 100%;
        border-radius: 20px;
        background-color: #4CAF50;
        color: white;
    }
</style>
""", unsafe_allow_html=True) # <--- MAKE SURE THIS SAYS html

# 3. Sidebar for Project Info
with st.sidebar:
    st.title("📁 Project Info")
    st.info("***System:** Vision-Language Model")
    st.info("***Model:** BLIP (Salesforce)")
    st.info("***Framework:** Streamlit + PyTorch")
    st.divider()
    st.write("Created by: [Your Name]")

# 4. Main Header
st.title("🔥 AI Image to Prompt Generator")
st.write("Transform your images into high-quality descriptive prompts for Stable Diffusion & Midjourney.")

# 5. Layout Columns
col1, col2 = st.columns([1, 1])

with col1:
    st.subheader("🖼️ Input Image")
    uploaded_file = st.file_uploader("Drag and drop your image here", type=["jpg", "png", "jpeg"])
    if uploaded_file:
        image = Image.open(uploaded_file).convert('RGB')
        st.image(image, use_container_width=True)

with col2:
    st.subheader("🖼️ Generated Output")
    if uploaded_file:
        if st.button('🚀 Analyze & Generate Prompt'):
            with st.spinner('AI is analyzing features...'):
                # Load Model
                processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
                model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base").to("cuda")

                # Process
                inputs = processor(image, return_tensors="pt").to("cuda")
                out = model.generate(**inputs)
                caption = processor.decode(out[0], skip_special_tokens=True)

                # Display Results in attractive boxes
                st.markdown("### 🔥 Optimized Prompt")
                st.success(f"{caption}, highly detailed, 8k resolution, cinematic lighting, masterpiece, sharp focus, digital")

                st.markdown("### 🚫 Negative Prompt")
                st.warning("blurry, low quality, distorted, watermark, text, grainy, extra fingers, out of frame")
            else:
                st.write("Please upload an image to see the generated results.")
    
```

Overwriting app.py

```
!streamlit run app.py & npx localtunnel --port 8501
```

```
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.
```

```

    7: ::ffff:192.168.1.100: your url is: https://cute-pots-attack.local.lt

```

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>
Network URL: <http://172.28.0.12:8501>
External URL: <http://34.125.101.94:8501>

Stopping...

 $\wedge C$

```
!streamlit run app.py & npx localtunnel --port 8501
```

```

"""
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.

"""
your url is: https://small-eagles-pay.loca.lt

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.28.0.12:8501
External URL: http://34.125.101.94:8501

2025-12-24 06:22:28.994324: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: A
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1766557349.028584 8167 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attempting to register factory for p
E0000 00:00:1766557349.039131 8167 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attempting to register factory for p
W0000 00:00:1766557349.064917 8167 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1766557349.064953 8167 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1766557349.064961 8167 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1766557349.064967 8167 computation_placer.cc:177] computation placer already registered. Please check linkage a
2025-12-24 06:22:29.072478: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use av
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler
2025-12-24 06:25:08.801 Please replace `use_container_width` with `width`.

`use_container_width` will be removed after 2025-12-31.

For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
2025-12-24 06:25:22.427 Please replace `use_container_width` with `width`.

`use_container_width` will be removed after 2025-12-31.

For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
Using a slow image processor as `use_fast` is unset and a slow processor was saved with this model. `use_fast=True` will be th
preprocessor_config.json: 100% 287/287 [00:00<00:00, 2.65MB/s]
tokenizer_config.json: 100% 506/506 [00:00<00:00, 4.98MB/s]
vocab.txt: 232kB [00:00, 18.3MB/s]
tokenizer.json: 711kB [00:00, 57.2MB/s]
special_tokens_map.json: 100% 125/125 [00:00<00:00, 1.46MB/s]
config.json: 4.56kB [00:00, 28.5MB/s]
pytorch_model.bin: 100% 990M/990M [00:07<00:00, 134MB/s]
model.safetensors: 100% 990M/990M [00:08<00:00, 111MB/s]
Stopping...
^C

```