

```
!pip install opencv-python rembg==2.0.66 onnxruntime
```

```

Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: rembg==2.0.66 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: onnxruntime in /usr/local/lib/python3.11/dist-packages (1
Requirement already satisfied: jsonschema in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from re
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (from r
Requirement already satisfied: pooch in /usr/local/lib/python3.11/dist-packages (from re
Requirement already satisfied: pymatting in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: scikit-image in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from re
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from ren
Requirement already satisfied: coloredlogs in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: protobuf in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: sympy in /usr/local/lib/python3.11/dist-packages (from or
Requirement already satisfied: humanfriendly>=9.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/py
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: numba!=0.49.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: networkx>=3.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.11/c
Requirement already satisfied: typing-extensions>=4.4.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dis
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-pack

```

```

import rembg
import cv2
import numpy as np
from rembg import remove
from google.colab import files
from google.colab.patches import cv2_imshow
from PIL import Image
import io

# Upload an image
uploaded = files.upload()

```

```
# Get the filename
for filename in uploaded.keys():
    img_filename = filename

# Read the image using PIL to get bytes
with open(img_filename, "rb") as f:
    input_bytes = f.read()

# Remove background
output_bytes = remove(input_bytes)

# Convert result back to image for display/saving
output_image = Image.open(io.BytesIO(output_bytes)).convert("RGBA")
output_np = np.array(output_image)

# Convert RGBA to BGRA for OpenCV
output_np_bgra = cv2.cvtColor(output_np, cv2.COLOR_RGBA2BGRA)

# Display the image
cv2_imshow(output_np_bgra)

# Save the output image
output_image.save("output_image_no_background.png")
print("Image saved to output_image_no_background.png")
```



Choose Files

No file chosen

Upload widget is only available when the cell has been executed in this cell to enable.

Downloading data from '<https://github.com/danielgatis/rembg/releases/download/v0.0.0/u2r>

Saving dog 3.jpg to dog 3 (1).jpg

[illegible]

```
!pip install demucs pydub
```



```
Requirement already satisfied: demucs in /usr/local/lib/python3.11/dist-packages (4.0.1)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (0.25.1)
Requirement already satisfied: dora-search in /usr/local/lib/python3.11/dist-packages (4.0.1)
Requirement already satisfied: einops in /usr/local/lib/python3.11/dist-packages (from dora-search) (0.7.0)
Requirement already satisfied: julius>=0.2.3 in /usr/local/lib/python3.11/dist-packages (from dora-search) (0.2.3)
Requirement already satisfied: lameenc>=1.2 in /usr/local/lib/python3.11/dist-packages (from dora-search) (1.2.0)
Requirement already satisfied: openunmix in /usr/local/lib/python3.11/dist-packages (from dora-search) (0.1.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from dora-search) (6.0.1)
Requirement already satisfied: torch>=1.8.1 in /usr/local/lib/python3.11/dist-packages (from dora-search) (2.2.2)
Requirement already satisfied: torchaudio>=0.8 in /usr/local/lib/python3.11/dist-packages (from dora-search) (2.2.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from dora-search) (4.66.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from torch) (3.13.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11/dist-packages (from torch) (4.11.0)
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from torch) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from torch) (2024.9.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch) (12.4.127)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch) (12.4.127)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch) (12.4.127)
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/python3.11/dist-packages (from torch) (9.1.0.70)
```

```
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/python3.11/
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/python3.11/
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/lib/python3.
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/lib/python3.
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.1
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dis
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/c
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/local/lib/python3
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: omegaconf in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: retrying in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: submitit in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: treetable in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from op
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: antlr4-python3-runtime==4.9.* in /usr/local/lib/python3.1
Requirement already satisfied: six>=1.7.0 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: cloudpickle>=1.2.1 in /usr/local/lib/python3.11/dist-pack
```

```
from google.colab import files
uploaded = files.upload()
```

```
# Get the filename
for filename in uploaded.keys():
    audio_filename = filename
```



Choose Files No file chosen

Upload widget is only available when the cell has been executed in this cell to enable.

Saving [iSongs.info] 01 - War 2 Teaser.mp3 to [iSongs.info] 01 - War 2 Teaser.mp3

```
!python -m demucs --two-stems=vocals "{audio_filename}"
```



Important: the default model was recently changed to `htdemucs` the latest Hybrid Transf

Selected model is a bag of 1 models. You will see that many progress bars per track.

Separated tracks will be stored in `/content/separated/htdemucs`

```
Separating track [iSongs.info] 01 - War 2 Teaser.mp3
```

100% | 99.44999999999999/99.44999999999999

```
from pydub import AudioSegment
from IPython.display import Audio
import os
```

```
# Assuming the original filename was 'your_song.wav' or 'your_song.mp3'
# The separated files will be in 'separated/htdemucs/your_song'
# The music track will be named 'no_vocals.wav' or 'no_vocals.mp3'
# The vocals track will be named 'vocals.wav' or 'vocals.mp3'
```

```

# Example for playing the music track:
# Replace 'your_song' with the actual name of your uploaded file (without the ext
# and adjust the extension (.wav or .mp3) if necessary.

# Get the base name of the uploaded file
base_filename = audio_filename.rsplit('.', 1)[0]
# audio_extension = audio_filename.rsplit('.', 1)[1] # We will determine the acti

# Construct the base directory path
output_dir = f"separated/htdemucs/{base_filename}"

# Check contents of the output directory
print(f"Contents of {output_dir}:")
try:
    for item in os.listdir(output_dir):
        print(item)
except FileNotFoundError:
    print(f"Directory {output_dir} not found. Demucs may have failed.")

# Try to find the music track file, assuming .wav or .mp3 output
music_track_path_wav = f"{output_dir}/no_vocals.wav"
music_track_path_mp3 = f"{output_dir}/no_vocals.mp3"

music_track_path = None
if os.path.exists(music_track_path_wav):
    music_track_path = music_track_path_wav
    audio_extension = "wav"
elif os.path.exists(music_track_path_mp3):
    music_track_path = music_track_path_mp3
    audio_extension = "mp3"
else:
    print("Could not find 'no_vocals.wav' or 'no_vocals.mp3' in the output direct

if music_track_path:
    print(f"Playing the music track from: {music_track_path}")
    try:
        music_track = AudioSegment.from_file(music_track_path)
        display(Audio(music_track.export(format=audio_extension).read()))
    except Exception as e:
        print(f"Error loading or playing audio file: {e}")

# You can do the same for the vocals track:
# vocals_track_path_wav = f"{output_dir}/vocals.wav"
# vocals_track_path_mp3 = f"{output_dir}/vocals.mp3"
#
# vocals_track_path = None
# if os.path.exists(vocals_track_path_wav):
#     vocals_track_path = vocals_track_path_wav
# elif os.path.exists(vocals_track_path_mp3):
#     vocals_track_path = vocals_track_path_mp3
#

```

```
# if vocals_track_path:
#     print("Playing the vocals track:")
#     vocals_track = AudioSegment.from_file(vocals_track_path)
#     display(Audio(vocals_track.export(format=audio_extension).read()))
```



Contents of separated/htdemucs/[iSongs.info] 01 - War 2 Teaser:

vocals.wav

no_vocals.wav

Playing the music track from: separated/htdemucs/[iSongs.info] 01 - War 2 Teaser/no_vocals.wav

0:00 / 1:34

```
from google.colab import files
import os
```

```
# Replace 'your_song' with the actual name of your uploaded file (without the extension)
# and adjust the extension (.wav or .mp3) if necessary.
```

```
# Get the base name of the uploaded file
base_filename = audio_filename.rsplit('.', 1)[0]
```

```
# Download the music track
music_track_path_wav = f"separated/htdemucs/{base_filename}/no_vocals.wav"
music_track_path_mp3 = f"separated/htdemucs/{base_filename}/no_vocals.mp3"
```

```
downloaded_music = False
if os.path.exists(music_track_path_wav):
    try:
        files.download(music_track_path_wav)
        print(f"Downloaded {music_track_path_wav}")
        downloaded_music = True
    except Exception as e:
        print(f"Error downloading music track (.wav): {e}")
elif os.path.exists(music_track_path_mp3):
    try:
        files.download(music_track_path_mp3)
        print(f"Downloaded {music_track_path_mp3}")
        downloaded_music = True
    except Exception as e:
        print(f"Error downloading music track (.mp3): {e}")
```

```
if not downloaded_music:
    print(f"Could not find or download music track (.wav or .mp3) in separated/htdemucs/{base_filename}")
```


```
# Download the vocals track
vocals_track_path_wav = f"separated/htdemucs/{base_filename}/vocals.wav"
vocals_track_path_mp3 = f"separated/htdemucs/{base_filename}/vocals.mp3"
```

```

downloaded_vocals = False
if os.path.exists(vocals_track_path_wav):
    try:
        files.download(vocals_track_path_wav)
        print(f"Downloaded {vocals_track_path_wav}")
        downloaded_vocals = True
    except Exception as e:
        print(f"Error downloading vocals track (.wav): {e}")
elif os.path.exists(vocals_track_path_mp3):
    try:
        files.download(vocals_track_path_mp3)
        print(f"Downloaded {vocals_track_path_mp3}")
        downloaded_vocals = True
    except Exception as e:
        print(f"Error downloading vocals track (.mp3): {e}")

if not downloaded_vocals:
    print(f"Could not find or download vocals track (.wav or .mp3) in separated/htdemucs/{base_filename}")

```

 Downloaded separated/htdemucs/[iSongs.info] 01 - War 2 Teaser/no_vocals.wav
 Downloaded separated/htdemucs/[iSongs.info] 01 - War 2 Teaser/vocals.wav

```

!pip install demucs pydub
from google.colab import files
uploaded = files.upload()

# Get the filename
for filename in uploaded.keys():
    audio_filename = filename
!python -m demucs --two-stems=vocals "{audio_filename}"
from pydub import AudioSegment
from IPython.display import Audio
import os

# Assuming the original filename was 'your_song.wav' or 'your_song.mp3'
# The separated files will be in 'separated/htdemucs/your_song'
# The music track will be named 'no_vocals.wav' or 'no_vocals.mp3'
# The vocals track will be named 'vocals.wav' or 'vocals.mp3'

# Example for playing the music track:
# Replace 'your_song' with the actual name of your uploaded file (without the ext
# and adjust the extension (.wav or .mp3) if necessary.

# Get the base name of the uploaded file
base_filename = audio_filename.rsplit('.', 1)[0]
# audio_extension = audio_filename.rsplit('.', 1)[1] # We will determine the actual extension

# Construct the base directory path
output_dir = f"separated/htdemucs/{base_filename}"

# Check contents of the output directory

```

```

print(f"Contents of {output_dir}:")
try:
    for item in os.listdir(output_dir):
        print(item)
except FileNotFoundError:
    print(f"Directory {output_dir} not found. Demucs may have failed.")

# Try to find the music track file, assuming .wav or .mp3 output
music_track_path_wav = f"{output_dir}/no_vocals.wav"
music_track_path_mp3 = f"{output_dir}/no_vocals.mp3"

music_track_path = None
if os.path.exists(music_track_path_wav):
    music_track_path = music_track_path_wav
    audio_extension = "wav"
elif os.path.exists(music_track_path_mp3):
    music_track_path = music_track_path_mp3
    audio_extension = "mp3"
else:
    print("Could not find 'no_vocals.wav' or 'no_vocals.mp3' in the output direct

if music_track_path:
    print(f"Playing the music track from: {music_track_path}")
    try:
        music_track = AudioSegment.from_file(music_track_path)
        display(Audio(music_track.export(format=audio_extension).read()))
    except Exception as e:
        print(f"Error loading or playing audio file: {e}")

# You can do the same for the vocals track:
# vocals_track_path_wav = f"{output_dir}/vocals.wav"
# vocals_track_path_mp3 = f"{output_dir}/vocals.mp3"
#
# vocals_track_path = None
# if os.path.exists(vocals_track_path_wav):
#     vocals_track_path = vocals_track_path_wav
# elif os.path.exists(vocals_track_path_mp3):
#     vocals_track_path = vocals_track_path_mp3
#
# if vocals_track_path:
#     print("Playing the vocals track:")
#     vocals_track = AudioSegment.from_file(vocals_track_path)
#     display(Audio(vocals_track.export(format=audio_extension).read()))

from google.colab import files
import os

# Replace 'your_song' with the actual name of your uploaded file (without the ext
# and adjust the extension (.wav or .mp3) if necessary.

# Get the base name of the uploaded file
base_filename = audio_filename.rsplit('.', 1)[0]

```



```
# Download the vocals track
vocals_track_path_wav = f"separated/htdemucs/{base_filename}/vocals.wav"
vocals_track_path_mp3 = f"separated/htdemucs/{base_filename}/vocals.mp3"

downloaded_vocals = False
if os.path.exists(vocals_track_path_wav):
    try:
        files.download(vocals_track_path_wav)
        print(f"Downloaded {vocals_track_path_wav}")
        downloaded_vocals = True
    except Exception as e:
        print(f"Error downloading vocals track (.wav): {e}")
elif os.path.exists(vocals_track_path_mp3):
    try:
        files.download(vocals_track_path_mp3)
        print(f"Downloaded {vocals_track_path_mp3}")
        downloaded_vocals = True
    except Exception as e:
        print(f"Error downloading vocals track (.mp3): {e}")

if not downloaded_vocals:
    print(f"Could not find or download vocals track (.wav or .mp3) in separated/
```



Requirement already satisfied: demucs in /usr/local/lib/python3.11/dist-packages (4.0.1)
 Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (0.25.1)
 Requirement already satisfied: dora-search in /usr/local/lib/python3.11/dist-packages (1.0.0)
 Requirement already satisfied: einops in /usr/local/lib/python3.11/dist-packages (from c
 Requirement already satisfied: julius>=0.2.3 in /usr/local/lib/python3.11/dist-packages
 Requirement already satisfied: lameenc>=1.2 in /usr/local/lib/python3.11/dist-packages (f
 Requirement already satisfied: openunmix in /usr/local/lib/python3.11/dist-packages (fro
 Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from c
 Requirement already satisfied: torch>=1.8.1 in /usr/local/lib/python3.11/dist-packages (f
 Requirement already satisfied: torchaudio>=0.8 in /usr/local/lib/python3.11/dist-package
 Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from den
 Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from
 Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11/di
 Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from
 Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from t
 Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from t
 Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/lib/pythor
 Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/pyth
 Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/lib/pythor
 Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/python3.11/
 Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/python3.11/
 Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/python3.11/
 Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/lib/python3.
 Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/lib/python3.
 Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in /usr/local/lib/pythor
 Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.1
 Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dis
 Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/