

```
!pip install transformers gradio tensorflow emoji
```

```
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.37.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->tensorflow) (0.43.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.4.26)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (0.14.0)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.0.9)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.15.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.10.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.0)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (3.7.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (0.17.0)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (3.0.6)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorflow) (2.18.0)
Requirement already satisfied: mdurl>=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow) (0.1.2)
Downloading gradio-5.29.0-py3-none-any.whl (54.1 MB)
54.1/54.1 MB 15.1 MB/s eta 0:00:00
Downloading gradio_client-1.10.0-py3-none-any.whl (322 kB)
322.9/322.9 kB 17.0 MB/s eta 0:00:00
Downloading emoji-2.14.1-py3-none-any.whl (590 kB)
590.6/590.6 kB 28.2 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
95.2/95.2 kB 5.5 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloading ruff-0.11.8-py3-none-manylinux_2_17_x86_64_musl_x86_64.whl (11.5 MB)
11.5/11.5 MB 67.0 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
72.0/72.0 kB 3.1 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Downloading uvicorn-0.34.2-py3-none-any.whl (62 kB)
62.5/62.5 kB 3.7 MB/s eta 0:00:00
Downloading ffmpeg-0.5.0-py3-none-any.whl (6.0 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub, uvicorn, tomlkit, semantic-version, ruff, python-multipart, groovy, ffmpeg, emoji, aiofiles
Successfully installed aiofiles-24.1.0 emoji-2.14.1 fastapi-0.115.12 ffmpeg-0.5.0 gradio-5.29.0 gradio-client-1.10.0 groovy-0.1.2
```

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

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving archive (2).zip to archive (2).zip
 Saving ReactionGIF.ids.json to ReactionGIF.ids.json
 Saving face_emotion.h5 to face_emotion.h5
 Saving emotiontxt.txt to emotiontxt.txt

```
import os
print("face_emotion.h5" in os.listdir())
```

True

```

from transformers import pipeline
import tensorflow as tf
from tensorflow.keras.models import load_model
import gradio as gr
import emoji, json, re, numpy as np
from PIL import Image

# Load pretrained DistilBERT for text sentiment analysis:contentReference[oaicite:5]{index=5}
text_analyzer = pipeline("sentiment-analysis", model="distilbert-base-uncased-finetuned-sst-2-english")

```

🔄 Device set to use cpu

```
!pip install keras opencv-python
```

🔄 Requirement already satisfied: keras in /usr/local/lib/python3.11/dist-packages (3.8.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: absl-py in /usr/local/lib/python3.11/dist-packages (from keras) (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from keras) (2.0.2)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras) (0.0.9)
Requirement already satisfied: h5py in /usr/local/lib/python3.11/dist-packages (from keras) (3.13.0)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras) (0.15.0)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.11/dist-packages (from keras) (0.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from keras) (24.2)
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.11/dist-packages (from optree->keras) (4.13.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras) (2.19.1)
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras)

```
# Define a mapping from specific emoji characters to basic sentiments
```

```

emoji_sentiment_map = {
    "😊": "happy", "😄": "happy", "😁": "happy", "😃": "happy",
    "😅": "happy", "😆": "happy", "😂": "happy", "😇": "happy", "😍": "happy",
    "😬": "happy", "😏": "happy",
    "😞": "sad", "😓": "sad", "😔": "sad", "😕": "sad",
    "😖": "sad", "😗": "sad", "😘": "sad", "😙": "sad", "😚": "sad",
    "😛": "sad", "😜": "sad", "😝": "sad", "😞": "sad", "😟": "sad",
    "😠": "angry", "😡": "angry", "😢": "angry", "😣": "angry",
    "😤": "angry"
}

```

```

import os
import json

```

```

# Confirm the correct path
file_path = "ReactionGIF.ids.json"

```

```

if os.path.exists(file_path):
    gif_sentiment_map = {}
    with open(file_path, "r", encoding="utf-8") as f:
        for line in f:
            entry = json.loads(line)
            gif_sentiment_map[str(entry["original_id"])] = entry["label"]
    print("GIF sentiment map loaded successfully.")
else:
    print("File not found. Please ensure 'ReactionGIF.ids.json' is uploaded to the same directory.")

```

🔄 GIF sentiment map loaded successfully.

```

def analyze(text, face_img, gif_id, emoji_img):
    # 1. Text sentiment with DistilBERT
    if text:
        res = text_analyzer(text)[0]
        sent = res['label'].capitalize()
        score = res['score'] * 100
        text_result = f"Text Sentiment: {sent} ({score:.1f}%)"
    else:
        text_result = "Text Sentiment: (no text provided)"

    # 2. Face image emotion
    if face_img:
        face_gray = face_img.convert('L').resize((48,48))
        arr = np.array(face_gray) / 255.0
        arr = arr.reshape((1,48,48,1))
        pred = face_model.predict(arr)
        label = face_labels[np.argmax(pred)]
        face_result = f"Face Emotion: {label}"
    else:
        face_result = "Face Emotion: (no image)"

    # 3. Emoji sentiment (from text)

```

```

emojis_found = emoji.emoji_list(text) if text else []
if emojis_found:
    sentiments = set()
    for item in emojis_found:
        char = item['emoji']
        sentiments.add(emoji_sentiment_map.get(char, "neutral"))
    emoji_sent = ", ".join(sorted(sentiments))
    emoji_result = f"Emoji Sentiment: {emoji_sent}"
else:
    emoji_result = "Emoji Sentiment: (no emojis found)"


# 4. GIF ID sentiment (yes → Positive, no → Negative)
if gif_id:
    label = gif_sentiment_map.get(gif_id)
    if label:
        gif_result = f"GIF Sentiment: {'Positive' if label=='yes' else 'Negative'}"
    else:
        gif_result = "GIF Sentiment: (ID not found)"
else:
    gif_result = "GIF Sentiment: (no ID provided)"

return text_result, face_result, emoji_result, gif_result

iface = gr.Interface(
    fn=analyze,
    inputs=[
        gr.Textbox(lines=2, label="Input Text (with emojis)"),
        gr.Image(type="pil", label="Face Image"),
        gr.Textbox(lines=1, label="Optional GIF Tweet ID"),
        gr.Image(type="pil", label="Optional Emoji Image")
    ],
    outputs=[
        gr.Textbox(label="Text Sentiment"),
        gr.Textbox(label="Face Emotion"),
        gr.Textbox(label="Emoji Sentiment"),
        gr.Textbox(label="GIF Sentiment")
    ],
    title="Multimodal Sentiment Analysis",
    description="Analyze sentiment from text, emojis, face, and GIF inputs.",
    theme=gr.themes.Monochrome() # uses a dark background with light text
)

iface.launch()

```

 It looks like you are running Gradio on a hosted Jupyter notebook. For the Gradio app to work, sharing must be enabled. Autor

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()

* Running on public URL: <https://3ab4668613459426cf.gradio.live>


This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the w

Multimodal Sentiment Analysis

Analyze sentiment from text, emojis, face, and GIF inputs.

Input Text (with emojis)

☒ Face Image



Drop Image Here
- or -
Click to Upload

```

from tensorflow.keras.models import load_model
import h5py

```

```
# Try to inspect the file
with h5py.File("face_emotion.h5", 'r') as f:
    print(list(f.keys()))
```

```
['conv2d', 'conv2d_1', 'conv2d_2', 'conv2d_3', 'dense', 'dense_1', 'dropout', 'dropout_1', 'dropout_2', 'flatten', 'max_pooling2d', 'max_pooling2d_1', 'max_pooling2d_2', 'top_level_model_weights']
```

```
import h5py
```

```
with h5py.File("face_emotion.h5", 'r') as f:
    def print_structure(name):
        print(name)
    f.visit(print_structure)
```

```
conv2d
conv2d/conv2d
conv2d/conv2d/bias:0
conv2d/conv2d/kernel:0
conv2d_1
conv2d_1/conv2d_1
conv2d_1/conv2d_1/bias:0
conv2d_1/conv2d_1/kernel:0
conv2d_2
conv2d_2/conv2d_2
conv2d_2/conv2d_2/bias:0
conv2d_2/conv2d_2/kernel:0
conv2d_3
conv2d_3/conv2d_3
conv2d_3/conv2d_3/bias:0
conv2d_3/conv2d_3/kernel:0
dense
dense/dense
dense/dense/bias:0
dense/dense/kernel:0
dense_1
dense_1/dense_1
dense_1/dense_1/bias:0
dense_1/dense_1/kernel:0
dropout
dropout_1
dropout_2
flatten
max_pooling2d
max_pooling2d_1
max_pooling2d_2
top_level_model_weights
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense
```

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(48, 48, 1)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),

    Flatten(),

    Dense(1024, activation='relu'),      # FIRST Dense layer
    Dense(7, activation='softmax')      # FINAL layer (7 classes)
])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape` argument to `Conv2D` or `Conv3D` layers. It is not used anymore.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

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

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
!pip install keras opencv-python
```

```

Requirement already satisfied: keras in /usr/local/lib/python3.11/dist-packages (3.8.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: absl-py in /usr/local/lib/python3.11/dist-packages (from keras) (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from keras) (2.0.2)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras) (0.0.9)
Requirement already satisfied: h5py in /usr/local/lib/python3.11/dist-packages (from keras) (3.13.0)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras) (0.15.0)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.11/dist-packages (from keras) (0.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from keras) (24.2)
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.11/dist-packages (from optree->keras) (4.13.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras)

```

```

from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense

```

```
# Define the model exactly as it was trained
```

```

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 1)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),

    Flatten(),
    Dense(1024, activation='relu'),
    Dense(6, activation='softmax') # FIXED: 6 classes instead of 7
])

```

```
# Load weights
```

```
model.load_weights("face_emotion.h5")
```

```
print("Model loaded successfully!")
```

```
# Define the model exactly as it was trained
```

```

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 1)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),

    Flatten(),
    Dense(1024, activation='relu'),
    Dense(6, activation='softmax') # FIXED: 6 classes instead of 7
])

```

```
# Load weights
```

```
model.load_weights("face_emotion.h5")
```

```
print("Model loaded successfully!")
```

```

Model loaded successfully!
Model loaded successfully!

```

```
!pip install -q gradio transformers
```

```

import gradio as gr
import numpy as np
import cv2
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense

```

```

# Emotion labels (6 classes)
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Neutral']

# Rebuild model architecture
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 1)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu'),

    Flatten(),
    Dense(1024, activation='relu'),
    Dense(6, activation='softmax') # MATCHING WEIGHTS
])

# Load pretrained weights
model.load_weights("face_emotion.h5")

# Preprocessing function
def process_image(img):
    if img is None:
        return "No image uploaded."

    # Convert to grayscale
    img_gray = cv2.cvtColor(np.array(img), cv2.COLOR_RGB2GRAY)
    img_resized = cv2.resize(img_gray, (64, 64))
    img_array = img_resized.reshape(1, 64, 64, 1).astype("float32") / 255.0

    # Predict
    prediction = model.predict(img_array)
    label = emotion_labels[np.argmax(prediction)]
    confidence = np.max(prediction)

    return f"{label} ({confidence*100:.2f}%)"

# Custom CSS for dark gold-black theme
css = """
body { background-color: black; color: white; }
.gradio-container {
    background-image: url('file/1000050514.jpg');
    background-size: cover;
    background-position: center;
    color: white;
}
"""

# Gradio interface
with gr.Blocks(css=css) as demo:
    gr.Markdown("<center><h1 style='color:gold;'>Sentiment Analysis - Face Emotion</h1></center>")

    with gr.Row():
        img = gr.Image(label="Upload Face Image", type="pil")

        out = gr.Textbox(label="Prediction Result")
        btn = gr.Button("Analyze Emotion")
        btn.click(process_image, inputs=img, outputs=out)

    gr.Markdown("<center><p style='color:white;'>Powered by CNN + Gradio UI</p></center>")

# Launch app
demo.launch()

```

```
↳ /usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape` to `Conv2D` or `Conv3D` layers. It is no longer necessary, and this usage will be deprecated. Instead, use the `input_shape` argument in the model's `compile` method.  
super().__init__(activity_regularizer=activity_regularizer, **kwargs)  
It looks like you are running Gradio on a hosted Jupyter notebook. For the Gradio app to work, sharing must be enabled. Auto
```


Colab notebook detected. To show errors in colab notebook, set debug=True in launch()




* Running on public URL: <https://e63c41e5c142aee72a.gradio.live>

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the w

Sentiment Analysis - Face Emotion

☒ Upload Face Image


Drop Image Here
- or -
Click to Upload

Prediction Result