```
!pip install numpy deepface fastapi opencv-python uvicorn pillow pyngrok
!pip install python-multipart
!ngrok authtoken 2tigyiCSpJnAnV82VnQniUearKM_45G7teJBzbbh2dWjpbxVX
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Collecting deepface
  Downloading deepface-0.0.93-py3-none-any.whl.metadata (30 kB)
Collecting fastapi
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Collecting uvicorn
  Downloading uvicorn-0.34.0-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (11.1.0)
Collecting pyngrok
  Downloading pyngrok-7.2.3-py3-none-any.whl.metadata (8.7 kB)
Requirement already satisfied: requests>=2.27.1 in /usr/local/lib/python3.11/dist-packages (from deepface) (2.32.3)
Requirement already satisfied: pandas>=0.23.4 in /usr/local/lib/python3.11/dist-packages (from deepface) (2.2.2)
Requirement already satisfied: gdown>=3.10.1 in /usr/local/lib/python3.11/dist-packages (from deepface) (5.2.0)
Requirement already satisfied: tqdm>=4.30.0 in /usr/local/lib/python3.11/dist-packages (from deepface) (4.67.1)
Requirement already satisfied: tensorflow>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from deepface) (2.18.0)
Requirement already satisfied: keras>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from deepface) (3.8.0)
Requirement already satisfied: Flask>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from deepface) (3.1.0)
Collecting flask-cors>=4.0.1 (from deepface)
  Downloading flask_cors-5.0.1-py3-none-any.whl.metadata (961 bytes)
Collecting mtcnn>=0.1.0 (from deepface)
  Downloading mtcnn-1.0.0-py3-none-any.whl.metadata (5.8 kB)
Collecting retina-face>=0.0.1 (from deepface)
  Downloading retina_face-0.0.17-py3-none-any.whl.metadata (10 kB)
Collecting fire>=0.4.0 (from deepface)
  Downloading fire-0.7.0.tar.gz (87 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 87.2/87.2 kB 3.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting gunicorn>=20.1.0 (from deepface)
  Downloading gunicorn-23.0.0-py3-none-any.whl.metadata (4.4 kB)
Collecting starlette<0.47.0,>=0.40.0 (from fastapi)
  Downloading starlette-0.46.1-py3-none-any.whl.metadata (6.2 kB)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.11/dist-packages (from fastapi) (4.12.2)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.11/dist-packages (from uvicorn) (8.1.8)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.11/dist-packages (from uvicorn) (0.14.0)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.11/dist-packages (from pyngrok) (6.0.2)
Requirement already satisfied: termcolor in /usr/local/lib/python3.11/dist-packages (from fire>=0.4.0->deepface) (2.5.0)
Requirement already satisfied: Werkzeug>=3.1 in /usr/local/lib/python3.11/dist-packages (from Flask>=1.1.2->deepface) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from Flask>=1.1.2->deepface) (3.1.6)
Requirement already satisfied: itsdangerous>=2.2 in /usr/local/lib/python3.11/dist-packages (from Flask>=1.1.2->deepface) (2.2.0)
Requirement already satisfied: blinker>=1.9 in /usr/local/lib/python3.11/dist-packages (from Flask>=1.1.2->deepface) (1.9.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from gdown>=3.10.1->deepface) (4.13.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from gdown>=3.10.1->deepface) (3.18.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from gunicorn>=20.1.0->deepface) (24.2)
Requirement already satisfied: absl-py in /usr/local/lib/python3.11/dist-packages (from keras>=2.2.0->deepface) (1.4.0)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=2.2.0->deepface) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=2.2.0->deepface) (0.0.8)
Requirement already satisfied: h5py in /usr/local/lib/python3.11/dist-packages (from keras>=2.2.0->deepface) (3.13.0)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=2.2.0->deepface) (0.14.1)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.11/dist-packages (from keras>=2.2.0->deepface) (0.4.1)
Requirement already satisfied: joblib>=1.4.2 in /usr/local/lib/python3.11/dist-packages (from mtcnn>=0.1.0->deepface) (1.4.2)
Collecting lz4>=4.3.3 (from mtcnn>=0.1.0->deepface)
  Downloading lz4-4.4.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.8 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.23.4->deepface)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.23.4->deepface) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.23.4->deepface) (2025.1)
```

```
import numpy as np
import cv2
from fastapi import FastAPI, File, UploadFile
from deepface import DeepFace
import uvicorn
from pyngrok import ngrok
from pathlib import Path
from PIL import Image
import io
import nest_asyncio
from fastapi.middleware.cors import CORSMiddleware
from fastapi.responses import JSONResponse

# Initialize FastAPI app
app = FastAPI()

# Enable CORS to allow frontend requests
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],  # Allow all origins, replace with frontend URL if needed
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

# Configure upload folder
UPLOAD_FOLDER = Path("uploads")
UPLOAD_FOLDER.mkdir(exist_ok=True)

# Emotion Context Dictionary
EMOTION_CONTEXT = {
    "angry": "The person appears to be angry. This could be due to frustration, conflict, or dissatisfaction.",
    "disgust": "The expression suggests disgust. This might be triggered by an unpleasant sight, smell, or thought.",
    "fear": "Fear is detected, which could be caused by a threat, danger, or overwhelming anxiety.",
    "happy": "A happy expression! The person might be experiencing joy, excitement, or a sense of achievement.",
    "sad": "The individual seems sad. Possible reasons could be personal loss, disappointment, or loneliness.",
    "surprise": "The person looks surprised, which could be due to an unexpected event, shock, or sudden realization.",
    "neutral": "A neutral expression. This could indicate calmness, focus, or a lack of strong emotions.",
}

# Emotion Colors
EMOTION_COLORS = {
    "angry": "#FF0000",
    "disgust": "#008000",
    "fear": "#800080",
    "happy": "#FFD700",
    "sad": "#0000FF",
    "surprise": "#FFA500",
    "neutral": "#808080",
}

# Emotion Emojis
EMOTION_EMOJIS = {
    "angry": "😡",
    "disgust": "🤢",
    "fear": "😨",
    "happy": "😃",
    "sad": "😢",
    "surprise": "😲",
    "neutral": "😐",
  }

# Function to determine intensity level
def get_intensity_level(percentage):
    if percentage < 50:
        return "Low 🟢"
    elif 50 <= percentage < 70:
        return "Normal 🟡"
    elif 70 <= percentage < 85:
        return "High 🟠"
    else:
        return "Extreme 🔴"

# Function to convert image file to NumPy array
def read_image_as_numpy(image_data):
    image = Image.open(io.BytesIO(image_data))
    image = image.convert("RGB")  # Convert to RGB
    img_np = np.array(image)  # Convert to NumPy array
    return img_np

# API endpoint for image upload and emotion detection
@app.post("/upload/")
async def upload_file(file: UploadFile = File(...)):
    try:
        # Read image as NumPy array
        image_data = await file.read()
        img_np = read_image_as_numpy(image_data)

        # Save the image temporarily
        filepath = UPLOAD_FOLDER / file.filename
        cv2.imwrite(str(filepath), cv2.cvtColor(img_np, cv2.COLOR_RGB2BGR))

        # Perform emotion recognition
        analysis = DeepFace.analyze(img_path=str(filepath), actions=["emotion"])

        # Extract dominant emotion and percentage
        emotions = analysis[0]["emotion"]

        # Convert all NumPy float32 to Python float
        emotions = {k: float(v) for k, v in emotions.items()}

        dominant_emotion = max(emotions, key=emotions.get)
        confidence = round(emotions[dominant_emotion], 2)

        # Get intensity level
        intensity = get_intensity_level(confidence)

        # Get context explanation
        emotion_context = EMOTION_CONTEXT.get(dominant_emotion, "Emotion not recognized.")

        # Create response
        response = {
            "emotion": dominant_emotion,
            "percentage": confidence,
            "emoji": EMOTION_EMOJIS.get(dominant_emotion, ""),
            "color": EMOTION_COLORS.get(dominant_emotion, "#000000"),
            "intensity": intensity,
            "context": emotion_context
        }

        return JSONResponse(content=response)

    except Exception as e:
        return JSONResponse(content={"error": str(e)})

# Start the ngrok tunnel and run FastAPI server
ngrok_tunnel = ngrok.connect(8000)
print("Public URL:", ngrok_tunnel.public_url)

nest_asyncio.apply()
uvicorn.run(app, host="0.0.0.0", port=8000)
```

```
25-03-24 08:09:58 - Directory /root/.deepface has been created
25-03-24 08:09:58 - Directory /root/.deepface/weights has been created
Public URL: https://18ea-35-185-55-176.ngrok-free.app
INFO:     Started server process [500]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
25-03-24 08:10:40 - facial_expression_model_weights.h5 will be downloaded...
Downloading...
From: https://github.com/serengil/deepface_models/releases/download/v1.0/facial_expression_model_weights.h5
To: /root/.deepface/weights/facial_expression_model_weights.h5
100%|██████████| 5.98M/5.98M [00:00<00:00, 70.2MB/s]
INFO:     27.116.21.222:0 - "POST /upload/ HTTP/1.1" 200 OK
INFO:     2401:4900:62d6:715b:60b3:8cb6:78ec:39fb:0 - "POST /upload/ HTTP/1.1" 200 OK
INFO:     2401:4900:62d6:715b:60b3:8cb6:78ec:39fb:0 - "POST /upload/ HTTP/1.1" 200 OK
```

Generate create a dataframe with 2 columns and 10 rows Close

```
from google.colab import auth
auth.authenticate_user()
```

```
!git config --global user.name "sathwikkumar1207"
!git config --global user.email "v.sathvikkumar7@gmail.com"
```

```
!git init
```

```
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /content/.git/
```

Generate create a dataframe with 2 columns and 10 rows Close

```
!git remote add orgin https://github.com/sathwikkumar1207/Smart-Emotion-Detection.git
```

```
!git remote -v
```

```
orgin   https://github.com/sathwikkumar1207/Smart-Emotion-Detection.git (fetch)
orgin   https://github.com/sathwikkumar1207/Smart-Emotion-Detection.git (push)
```

```
!git add .
!git commit -m "Intial commit"
!git branch -M main
```

```
!git push origin main
```

```
On branch main
nothing to commit, working tree clean
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

Start coding or generate with AI.