

# ?

# TABLE OF CONTENTS

- [1. Installing the required libraries](#)
  - [2. Emotion Classifier Taipy App for Prediction on User Images](#)
  - [3. Deployment of Emotion Classifier Taipy App on Render Platform](#)
    - [1. Steps for Deployment on Render Platform](#)
    - [2. Emotion Classifier Taipy App Python code](#)
    - [3. Requirements file](#)
    - [4. Storing Emotion Classifier Model in Git LFS](#)
    - [5. Failed Render Deployment](#)
  - [4. Emotion Classifier Gradio App for Prediction on User Images](#)
  - [5. Deployment for Real-time Prediction on Webcam Video](#)
  - [6. Model Retraining](#)
    - [1. Need for Model Retraining](#)
    - [2. When to do Model Retraining ?](#)
    - [3. How to do Model Retraining ?](#)
    - [4. A/B Testing](#)
- 

## 1. Installing the required libraries

```
In [1]: !pip3 install tensorflow keras numpy matplotlib opencv-python
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.25.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (4.8.0.76)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~0.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.
```

21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)  
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)  
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)  
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)  
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)  
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)  
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.0)  
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)  
Requirement already satisfied: tensorboard<2.16,>=2.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.2)  
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)  
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.53.0)  
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)  
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)  
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)  
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.43.0)  
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15->tensorflow) (2.27.0)  
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15->tensorflow) (1.2.0)  
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15->tensorflow) (3.6)  
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15->tensorflow) (2.31.0)  
Requirement already satisfied: tensorflow-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15->tensorflow) (0.7.2)  
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15->tensorflow) (3.0.3)  
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorflow<2.16,>=2.15->tensorflow) (5.3.3)  
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorflow<2.16,>=2.15->tensorflow) (0.4.0)  
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorflow<2.16,>=2.15->tensorflow) (4.9)  
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<2,>=0.5->tensorflow<2.16,>=2.15->tensorflow) (1.3.1)  
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow<2.16,>=2.15->tensorflow) (3.3.2)  
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow<2.16,>=2.15->tensorflow) (3.7)  
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow<2.16,>=2.15->tensorflow) (2.0.7)  
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow<2.16,>=2.15->tensorflow) (2024.6.2)  
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages

```
ges (from werkzeug>=1.0.1->tensorboard<2.16,>=2.15->tensorflow) (2.1.5)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tens orflow) (0.6.0)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorflow) (3.2.2)
```

---

## 2. Emotion Classifier Taipy App for Prediction on User Images

```
In [ ]: !pip3 install --ignore-installed taipy taipy-gui
```

```
Collecting taipy
  Downloading taipy-3.1.1-py3-none-any.whl (2.9 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.9/2.9 MB 10.2 MB/s eta 0:00:00
Collecting taipy-gui
  Downloading taipy-gui-3.1.3.tar.gz (2.7 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 22.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting cookiecutter<=2.5.0,>=2.1.1 (from taipy)
  Downloading cookiecutter-2.5.0-py3-none-any.whl (39 kB)
Collecting taipy-rest<3.2.0,>=3.1.1 (from taipy)
  Downloading taipy-rest-3.1.1.tar.gz (21 kB)
  Preparing metadata (setup.py) ... done
Collecting taipy-templates<3.2.0,>=3.1.1 (from taipy)
  Downloading taipy-templates-3.1.1.tar.gz (15 kB)
  Preparing metadata (setup.py) ... done
Collecting flask<=3.0.2,>=3.0.0 (from taipy-gui)
  Downloading flask-3.0.2-py3-none-any.whl (101 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.3/101.3 kB 11.7 MB/s eta 0:00:00
Collecting flask-cors<=4.0.0,>=4.0.0 (from taipy-gui)
  Downloading Flask_Cors-4.0.0-py2.py3-none-any.whl (14 kB)
Collecting flask-socketio<=5.3.6,>=5.3.6 (from taipy-gui)
  Downloading Flask_SocketIO-5.3.6-py3-none-any.whl (18 kB)
Collecting gevent<=23.9.1,>=23.7.0 (from taipy-gui)
  Downloading gevent-23.9.1-cp310-cp310-manylinux_2_28_x86_64.whl (6.4 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━ 6.4/6.4 MB 41.6 MB/s eta 0:00:00
Collecting gevent-websocket<=0.10.1,>=0.10.1 (from taipy-gui)
  Downloading gevent_websocket-0.10.1-py3-none-any.whl (22 kB)
Collecting gitignore-parser<=0.1.11,>=0.1 (from taipy-gui)
  Downloading gitignore_parser-0.1.11.tar.gz (5.3 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting kthread<=0.2.3,>=0.2.3 (from taipy-gui)
  Downloading kthread-0.2.3-py3-none-any.whl (3.9 kB)
Collecting markdown<=3.5.2,>=3.4.4 (from taipy-gui)
  Downloading Markdown-3.5.2-py3-none-any.whl (103 kB)
    ━━━━━━━━━━━━━━━━━━━━━━ 103.9/103.9 kB 13.6 MB/s eta 0:00:00
Collecting pandas<=2.2.0,>=1.3.5 (from taipy-gui)
  Downloading pandas-2.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1 3.0 MB)
    ━━━━━━━━━━━━━━━━━━━━━━ 13.0/13.0 MB 42.1 MB/s eta 0:00:00
Collecting python-dotenv<=1.0.1,>=1.0.0 (from taipy-gui)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Collecting pytz<=2023.3.post1,>=2021.3 (from taipy-gui)
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
    ━━━━━━━━━━━━━━━━━━━━ 502.5/502.5 kB 35.9 MB/s eta 0:00:00
Collecting simple-websocket<=1.0.0,>=0.10.1 (from taipy-gui)
```

```
  Downloading simple_websocket-1.0.0-py3-none-any.whl (13 kB)
Collecting taipy-config<3.2,>=3.1 (from taipy-gui)
  Downloading taipy-config-3.1.1.tar.gz (30 kB)
  Preparing metadata (setup.py) ... done
Collecting twisted<=23.10.0,>=23.8.0 (from taipy-gui)
  Downloading twisted-23.10.0-py3-none-any.whl (3.2 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 81.4 MB/s eta 0:00:00
Collecting tzlocal<=5.2,>=3.0 (from taipy-gui)
  Downloading tzlocal-5.2-py3-none-any.whl (17 kB)
Collecting binaryornot>=0.4.4 (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading binaryornot-0.4.4-py2.py3-none-any.whl (9.0 kB)
Collecting Jinja2<4.0.0,>=2.7 (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━ 133.3/133.3 kB 16.5 MB/s eta 0:00:00
Collecting click<9.0.0,>=7.0 (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
    ━━━━━━━━━━━━━━━━━━━ 97.9/97.9 kB 11.6 MB/s eta 0:00:00
Collecting pyyaml>=5.3.1 (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading PyYAML-6.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7
05 kB)
    ━━━━━━━━━━━━━━━ 705.5/705.5 kB 51.4 MB/s eta 0:00:00
Collecting python-slugify>=4.0.0 (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading python_slugify-8.0.4-py2.py3-none-any.whl (10 kB)
Collecting requests>=2.23.0 (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading requests-2.32.3-py3-none-any.whl (64 kB)
    ━━━━━━━ 64.9/64.9 kB 7.9 MB/s eta 0:00:00
Collecting arrow (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading arrow-1.3.0-py3-none-any.whl (66 kB)
    ━━━━━━━ 66.4/66.4 kB 8.3 MB/s eta 0:00:00
Collecting rich (from cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading rich-13.7.1-py3-none-any.whl (240 kB)
    ━━━━━━━ 240.7/240.7 kB 27.4 MB/s eta 0:00:00
Collecting Werkzeug>=3.0.0 (from flask<=3.0.2,>=3.0.0->taipy-gui)
  Downloading werkzeug-3.0.3-py3-none-any.whl (227 kB)
    ━━━━━━━ 227.3/227.3 kB 27.7 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2 (from flask<=3.0.2,>=3.0.0->taipy-gui)
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting blinker>=1.6.2 (from flask<=3.0.2,>=3.0.0->taipy-gui)
  Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Collecting python-socketio>=5.0.2 (from flask-socketio<=5.3.6,>=5.3.6->taipy-gui)
  Downloading python_socketio-5.11.3-py3-none-any.whl (76 kB)
    ━━━━━━━ 76.2/76.2 kB 10.1 MB/s eta 0:00:00
Collecting zope.event (from gevent<=23.9.1,>=23.7.0->taipy-gui)
  Downloading zope.event-5.0-py3-none-any.whl (6.8 kB)
Collecting zope.interface (from gevent<=23.9.1,>=23.7.0->taipy-gui)
  Downloading zope.interface-6.4.post2-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_6
4.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (247 kB)
    ━━━━━━━ 247.8/247.8 kB 26.8 MB/s eta 0:00:00
Collecting greenlet>=2.0.0 (from gevent<=23.9.1,>=23.7.0->taipy-gui)
  Downloading greenlet-3.0.3-cp310-cp310-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl
(616 kB)
    ━━━━━━━ 616.0/616.0 kB 44.8 MB/s eta 0:00:00
Collecting numpy<2,>=1.22.4 (from pandas<=2.2.0,>=1.3.5->taipy-gui)
  Downloading numpy-1.26.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1
8.2 MB)
    ━━━━━━━ 18.2/18.2 kB 53.4 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas<=2.2.0,>=1.3.5->taipy-gui)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
    ━━━━━━━ 229.9/229.9 kB 22.2 MB/s eta 0:00:00
Collecting tzdata>=2022.7 (from pandas<=2.2.0,>=1.3.5->taipy-gui)
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
    ━━━━━━━ 345.4/345.4 kB 32.8 MB/s eta 0:00:00
Collecting wsproto (from simple-websocket<=1.0.0,>=0.10.1->taipy-gui)
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Collecting deepdiff<=6.7.1,>=6.2.2 (from taipy-config<3.2,>=3.1->taipy-gui)
  Downloading deepdiff-6.7.1-py3-none-any.whl (76 kB)
```



```
Collecting charset-normalizer<4,>=2 (from requests>=2.23.0->cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading charset_normalizer-3.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (142 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 142.1/142.1 kB 19.1 MB/s eta 0:00:00
Collecting urllib3<3,>=1.21.1 (from requests>=2.23.0->cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading urllib3-2.2.2-py3-none-any.whl (121 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 121.4/121.4 kB 17.0 MB/s eta 0:00:00
Collecting certifi>=2017.4.17 (from requests>=2.23.0->cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading certifi-2024.6.2-py3-none-any.whl (164 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 164.4/164.4 kB 21.2 MB/s eta 0:00:00
Collecting boto3<=1.34.34,>=1.29.4 (from taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading boto3-1.34.34-py3-none-any.whl (139 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.3/139.3 kB 16.9 MB/s eta 0:00:00
Collecting networkx<=3.2.1,>=2.6 (from taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading networkx-3.2.1-py3-none-any.whl (1.6 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 71.6 MB/s eta 0:00:00
Collecting openpyxl<=3.1.2,>=3.1.2 (from taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━ 250.0/250.0 kB 29.0 MB/s eta 0:00:00
Collecting pyarrow<=15.0.0,>=14.0.2 (from taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading pyarrow-15.0.0-cp310-cp310-manylinux_2_28_x86_64.whl (38.3 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━ 38.3/38.3 MB 10.8 MB/s eta 0:00:00
Collecting pymongo[srv]<=4.6.1,>=4.2.0 (from taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading pymongo-4.6.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (677 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━ 677.1/677.1 kB 24.1 MB/s eta 0:00:00
Collecting sqlalchemy<=2.0.25,>=2.0.16 (from taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading SQLAlchemy-2.0.25-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 74.2 MB/s eta 0:00:00
Collecting setuptools (from zope.interface->gevent<=23.9.1,>=23.7.0->taipy-gui)
  Using cached setuptools-70.1.1-py3-none-any.whl (883 kB)
Collecting types-python-dateutil>=2.8.10 (from arrow->cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading types_python_dateutil-2.9.0.20240316-py3-none-any.whl (9.7 kB)
Collecting markdown-it-py>=2.2.0 (from rich->cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading markdown_it_py-3.0.0-py3-none-any.whl (87 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━ 87.5/87.5 kB 10.7 MB/s eta 0:00:00
Collecting pygments<3.0.0,>=2.13.0 (from rich->cookiecutter<=2.5.0,>=2.1.1->taipy)
  Downloading pygments-2.18.0-py3-none-any.whl (1.2 MB)
    ━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 55.6 MB/s eta 0:00:00
Collecting h11<1,>=0.9.0 (from wsproto->simple-websocket<=1.0.0,>=0.10.1->taipy-gui)
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
    ━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 7.5 MB/s eta 0:00:00
Collecting botocore<1.35.0,>=1.34.34 (from boto3<=1.34.34,>=1.29.4->taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading botocore-1.34.136-py3-none-any.whl (12.4 MB)
    ━━━━━━━━━━━━━━━━━━━━ 12.4/12.4 MB 57.8 MB/s eta 0:00:00
Collecting jmespath<2.0.0,>=0.7.1 (from boto3<=1.34.34,>=1.29.4->taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from boto3<=1.34.34,>=1.29.4->taipy-core<3.2.0,>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
  Downloading s3transfer-0.10.2-py3-none-any.whl (82 kB)
    ━━━━━━━━━━━━━━━━━━━━ 82.7/82.7 kB 10.7 MB/s eta 0:00:00
Collecting mdurl~0.1 (from markdown-it-py>=2.2.0->rich->cookiecutter<=2.5.0,>=2.1.1->taipy)
```

```
    Downloading mdurl-0.1.2-py3-none-any.whl (10.0 kB)
Collecting et-xmlfile (from openpyxl<=3.1.2,>=3.1.2->taipy-core<3.2.0,>=3.1.1->taipy-res
t<3.2.0,>=3.1.1->taipy)
    Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo[srv]<=4.6.1,>=4.2.0->taipy-core<3.2.0,
>=3.1.1->taipy-rest<3.2.0,>=3.1.1->taipy)
    Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
    00000000000000000000000000000000 307.7/307.7 kB 32.1 MB/s eta 0:00:00
Building wheels for collected packages: taipy-gui, gitignore-parser, taipy-config, taipy
-rest, taipy-templates, taipy-core
    Building wheel for taipy-gui (setup.py) ... done
    Created wheel for taipy-gui: filename=taipy_gui-3.1.3-py3-none-any.whl size=2793004 sh
a256=f92b88a1258f7feb090cf030e266996bcfb56b72e8e7be9424ee94b85cb093b2
    Stored in directory: /root/.cache/pip/wheels/d0/33/4c/f375272f39c50bdb153e6ecfd456bb7
a6fc36857659f259e1
    Building wheel for gitignore-parser (pyproject.toml) ... done
    Created wheel for gitignore-parser: filename=gitignore_parser-0.1.11-py3-none-any.whl size=4921 sha256=d48b6401ad87fde08b095ce7c4e3d9f1913319652064cff48d59a4b66315c099
    Stored in directory: /root/.cache/pip/wheels/64/c4/d8/fbeed21e95e39f02c89e10aade7f8072
d236f78d0c278121b0
    Building wheel for taipy-config (setup.py) ... done
    Created wheel for taipy-config: filename=taipy_config-3.1.1-py3-none-any.whl size=5527
2 sha256=177da5889d974db93f56a1641d5b5b6cfb732483a1f5f39936f1222bf166b0b0
    Stored in directory: /root/.cache/pip/wheels/d8/ee/4b/aa4febb3ebe4c80e1be1c6da3272e854
87aa0dfcbf1af00161
    Building wheel for taipy-rest (setup.py) ... done
    Created wheel for taipy-rest: filename=taipy_rest-3.1.1-py3-none-any.whl size=42759 sh
a256=07d9d9f60a153ca19db05309534b403f7576f69d4448e095da202a56eab4b98b
    Stored in directory: /root/.cache/pip/wheels/a0/8f/e8/5fa426883805c7b22932e62b22709265
8208f0f0cf857d2288
    Building wheel for taipy-templates (setup.py) ... done
    Created wheel for taipy-templates: filename=taipy_templates-3.1.1-py3-none-any.whl siz
e=33799 sha256=2d6c6a2715ec644a64ce01d0da25393633608b095015c367b4fb7747411752cd
    Stored in directory: /root/.cache/pip/wheels/50/ef/ee/10eef96dc1eaa6c28121a2a2a9b20cba
d1f01a84181cb04f8f
    Building wheel for taipy-core (setup.py) ... done
    Created wheel for taipy-core: filename=taipy_core-3.1.1-py3-none-any.whl size=277900 s
ha256=2ea803d4b7c1ea901e25011cbd2f1bc4d74a17e4321da9577aaf3092e8c4d98e
    Stored in directory: /root/.cache/pip/wheels/02/ec/4f/2cea0396333e546c7a7365b8fbf52627
10d6ed262cb483156b
Successfully built taipy-gui gitignore-parser taipy-config taipy-rest taipy-templates ta
ipy-core
Installing collected packages: text-unidecode, pytz, passlib, kthread, incremental, giti
gnore-parser, aniso8601, urllib3, tzlocal, tzdata, typing-extensions, types-python-dateu
til, toml, taipy-templates, six, setuptools, pyyaml, python-slugify, python-dotenv, pygm
ents, packaging, ordered-set, numpy, networkx, mdurl, MarkupSafe, markdown, jmespath, it
sdangerous, idna, h11, greenlet, et-xmlfile, dnspython, constantly, click, charset-norma
lizer, chardet, certifi, blinker, bidict, attrs, zope.interface, zope.event, wsproto, We
rkzeug, sqlalchemy, requests, python-dateutil, pymongo, pyarrow, openpyxl, marshmallow,
markdown-it-py, Jinja2, hyperlink, deepdiff, binaryornot, automat, apispec, twisted, ta
ipy-config, simple-websocket, rich, pandas, gevent, flask, botocore, arrow, s3transfer,
python-engineio, gevent-websocket, flask-restful, flask-cors, cookiecutter, apispec-web
frameworks, python-socketio, boto3, taipy-core, flask-socketio, taipy-rest, taipy-gui, t
aipy
```

```
ERROR: pip's dependency resolver does not currently take into account all the packages t
hat are installed. This behaviour is the source of the following dependency conflicts.
ipython 7.34.0 requires jedi>=0.16, which is not installed.
torch 2.3.0+cu121 requires nvidia-cublas-cu12==12.1.3.1; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cuda-cupti-cu12==12.1.105; platform_system == "Linux"
and platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cuda-nvrtc-cu12==12.1.105; platform_system == "Linux"
and platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cuda-runtime-cu12==12.1.105; platform_system == "Linu
x" and platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cudnn-cu12==8.9.2.26; platform_system == "Linux" and p
```

```
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cufft-cu12==11.0.2.54; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-curand-cu12==10.3.2.106; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cusolver-cu12==11.4.5.107; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-cusparse-cu12==12.1.0.106; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-nccl-cu12==2.20.5; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
torch 2.3.0+cu121 requires nvidia-nvtx-cu12==12.1.1.105; platform_system == "Linux" and
platform_machine == "x86_64", which is not installed.
cudf-cu12 24.4.1 requires pyarrow<15.0.0a0,>=14.0.1, but you have pyarrow 15.0.0 which is
incompatible.
google-colab 1.0.0 requires pandas==2.0.3, but you have pandas 2.2.0 which is incompatible.
google-colab 1.0.0 requires requests==2.31.0, but you have requests 2.32.3 which is incompatible.

Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.3 aniso8601-9.0.1 apispec-6.4.0 apispec-webframeworks-1.0.0 arrow-1.3.0 attrs-23.2.0 automat-22.10.0 bidict-0.23.1 binaryornot-0.4.4 blinker-1.8.2 boto3-1.34.34 botocore-1.34.136 certifi-2024.6.2 charset-normalizer-3.3.2 click-8.1.7 constantly-23.10.4 cookiecutter-2.5.0 dEEPDIFF-6.7.1 dnspython-2.6.1 et-xmlfile-1.1.0 flask-2.2.5 flask-cors-4.0.0 flask-restful-0.3.10 flask-socketio-5.3.6 gevent-23.9.1 gevent-websocket-0.10.1 gitignore-parser-0.1.11 greenlet-3.0.3 h11-0.14.0 hyperlink-21.0.0 idna-3.7 incremental-22.10.0 itsdangerous-2.2.0 jmespath-1.0.1 kthread-0.2.3 markdown-3.5.2 markdown-it-py-3.0.0 marshmallow-3.2.0.2 mdurl-0.1.2 networkx-3.2.1 numpy-1.25.2 openpyxl-3.1.2 ordered-set-4.1.0 packaging-2.4.1 pandas-2.0.3 passlib-1.7.4 pyarrow-14.0.2 pygments-2.16.1 pymongo-4.6.1 python-dateutil-2.8.2 python-dotenv-1.0.1 python-engineio-4.9.1 python-slugify-8.0.4 python-socketio-5.11.3 pytz-2023.3.post1 pyyaml-6.0.1 requests-2.31.0 rich-13.7.1 s3transfer-0.10.2 setuptools-67.7.2 simple-websocket-1.0.0 six-1.16.0 sqlalchemy-2.0.25 taipy-3.1.1 taipy-config-3.1.1 taipy-core-3.1.1 taipy-gui-3.1.3 taipy-rest-3.1.1 taipy-templates-3.1.1 text-unidecode-1.3 toml-0.10.2 twisted-23.10.0 types-python-dateutil-2.9.0.20240316 typing-extensions-4.12.2 tzdata-2024.1 tzlocal-5.2 urllib3-2.0.7 wsproto-1.2.0 zope.event-5.0.0 zope.interface-6.4.post2
```

```
In [ ]: # Imports
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import img_to_array
from taipy.gui import Gui, notify

In [ ]: # Mount Google Drive
from google.colab import drive
drive.mount('/gdrive')

In [ ]: # Upload the best ResNet50 model
model_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Vision/02"
model = tf.keras.models.load_model(model_path)

In [ ]: # Emotion labels
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']

In [ ]: # Initialize the face classifier
face_classifier_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer"
face_classifier = cv2.CascadeClassifier(face_classifier_path)

In [ ]: # Function to test Emotion Classifier on local images before deployment using Taipy GUI
def predict_emotion_from_path(image_path):
    """
    Predicts the emotion given the path to an image.
    """
```

```

Args:
    image_path (str): The path to the image file.

Returns:
    List of tuples: Each tuple contains the coordinates of the face (x, y, w, h) and
    """
# Load image
image = cv2.imread(image_path)
if image is None:
    raise ValueError(f"Image not found at path: {image_path}")

# Detect faces in the image
faces = face_classifier.detectMultiScale(image, scaleFactor=1.1, minNeighbors=5, min
emotions = []
for (x, y, w, h) in faces:
    face = image[y:y + h, x:x + w]
    face = cv2.resize(face, (224, 224))
    face = face.astype("float") / 255.0
    face = img_to_array(face)
    face = np.expand_dims(face, axis=0)

    prediction = model.predict(face)[0]
    emotion = emotion_labels[np.argmax(prediction)]
    emotions.append((x, y, w, h, emotion))
return emotions, image

```

```

In [ ]: import matplotlib.pyplot as plt

def display_image_with_emotions(image_path):
    emotions, image = predict_emotion_from_path(image_path)

    for (x, y, w, h, emotion) in emotions:
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(image, emotion, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255

    # Convert the image from BGR (OpenCV format) to RGB (Matplotlib format)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    plt.figure(figsize=(10, 10))
    plt.imshow(image)
    plt.axis('off') # Hide axes
    plt.show()

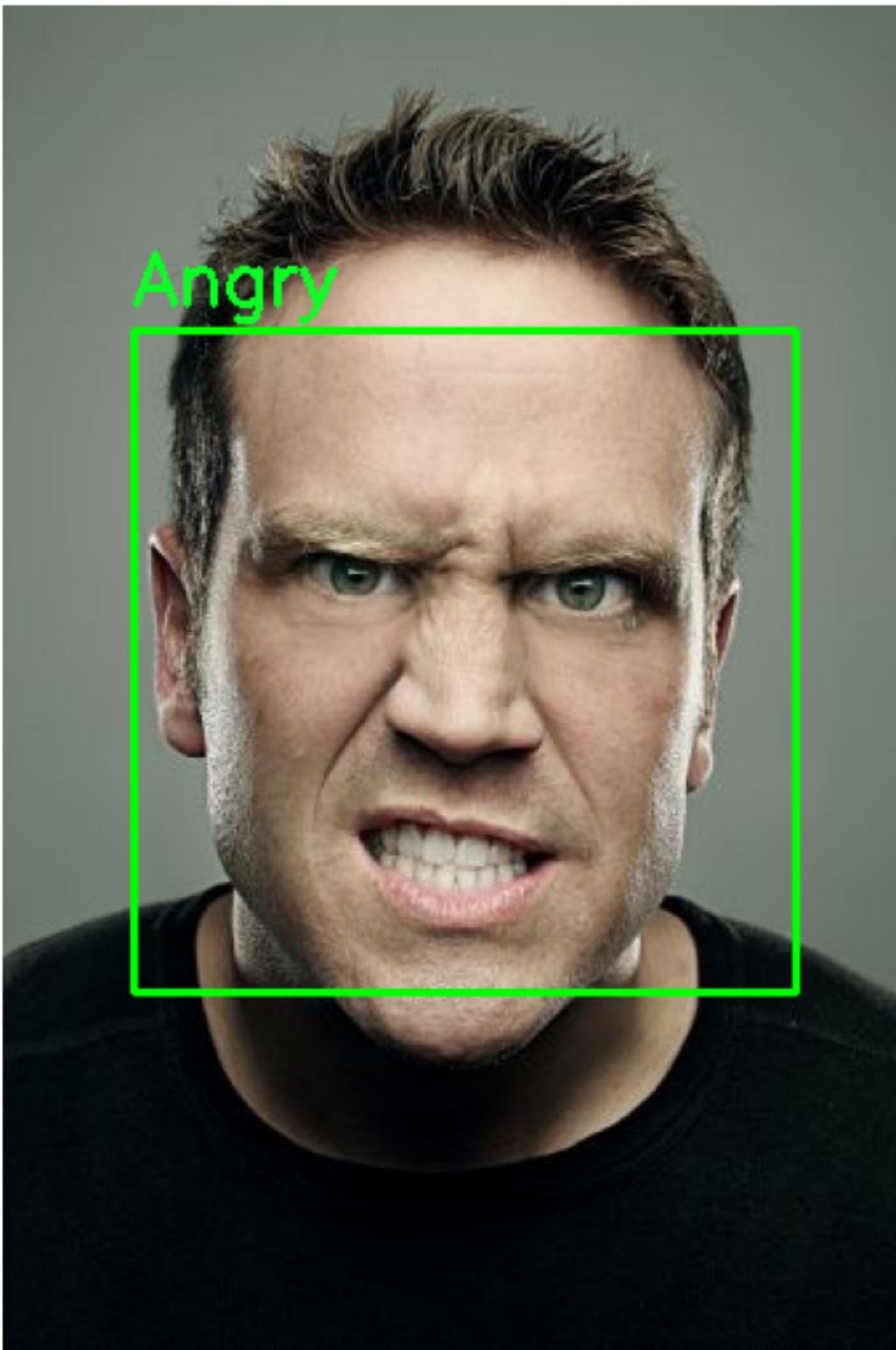
```

```

In [ ]: # Test "Angry" expression image
angry_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Vi
display_image_with_emotions(angry_image_path)

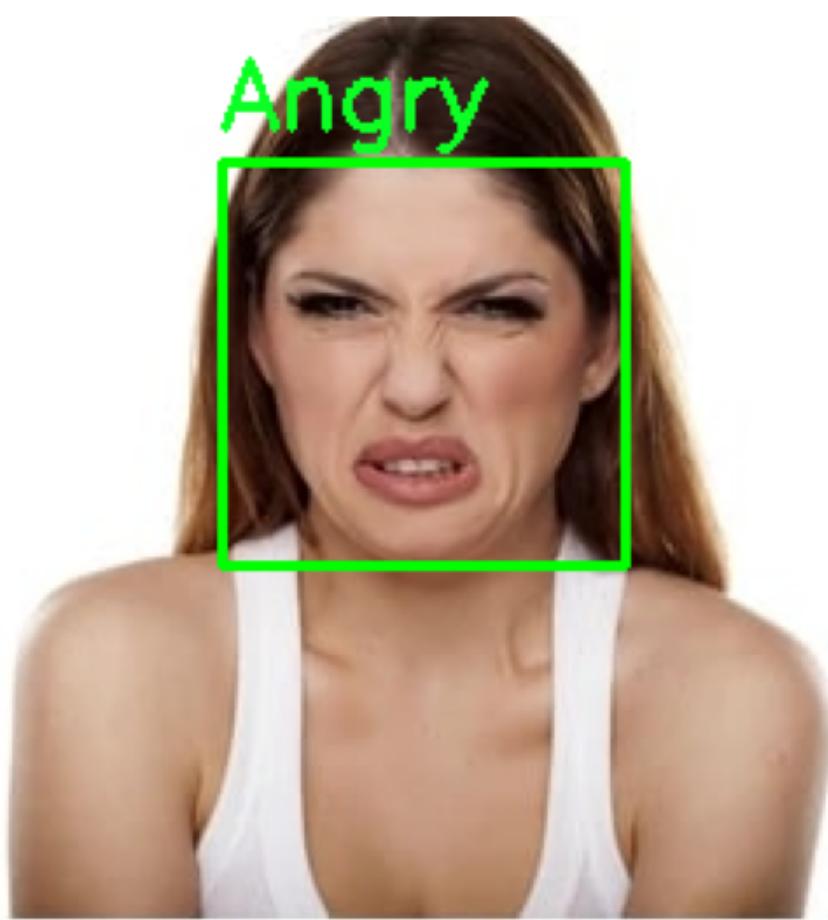
```

1/1 [=====] - 0s 322ms/step



```
In [ ]: # Test "Disgust" expression image
disgust_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-V
display_image_with_emotions(disgust_image_path)

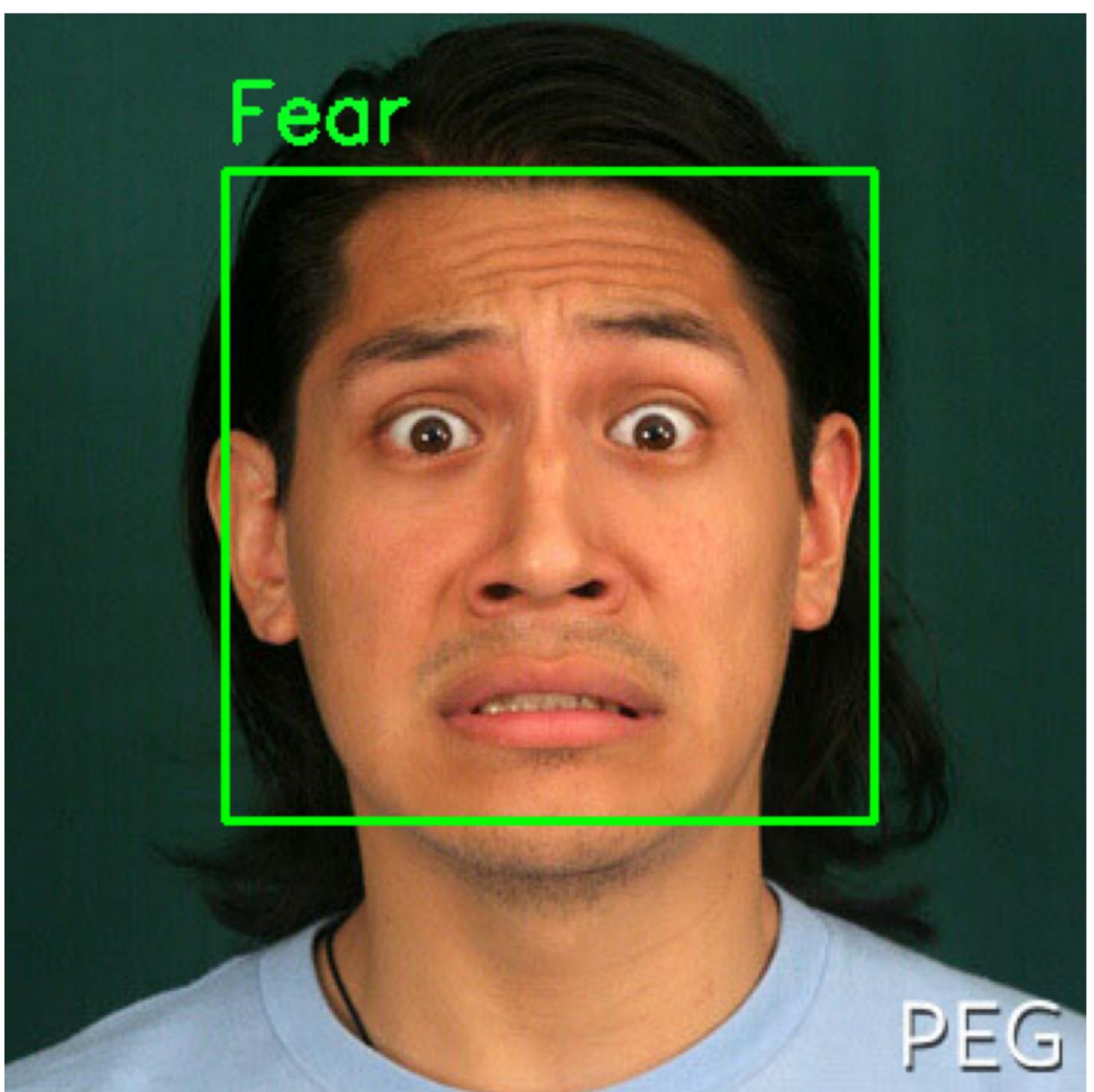
1/1 [=====] - 0s 183ms/step
```



shutterstock.com · 249074803

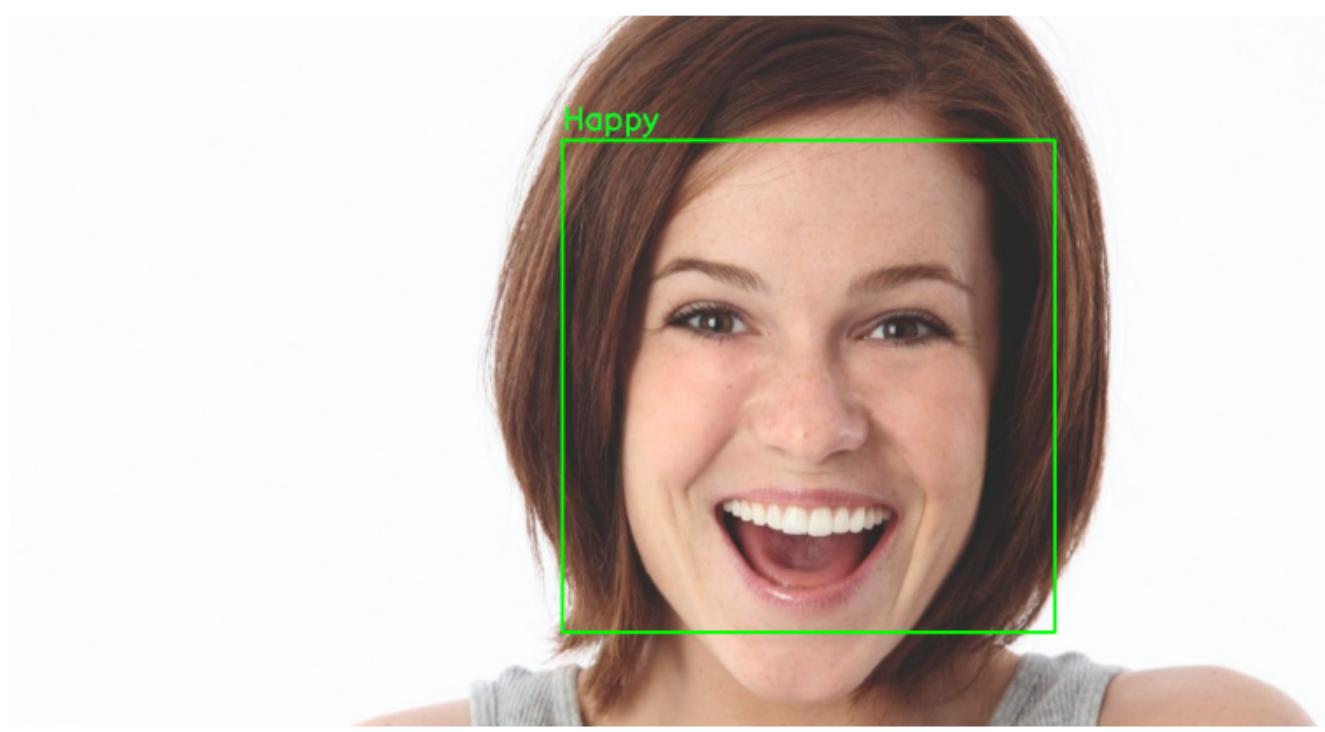
```
In [ ]: # Test "Fear" expression image
fear_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Visi
display_image_with_emotions(fear_image_path)

1/1 [=====] - 0s 185ms/step
```

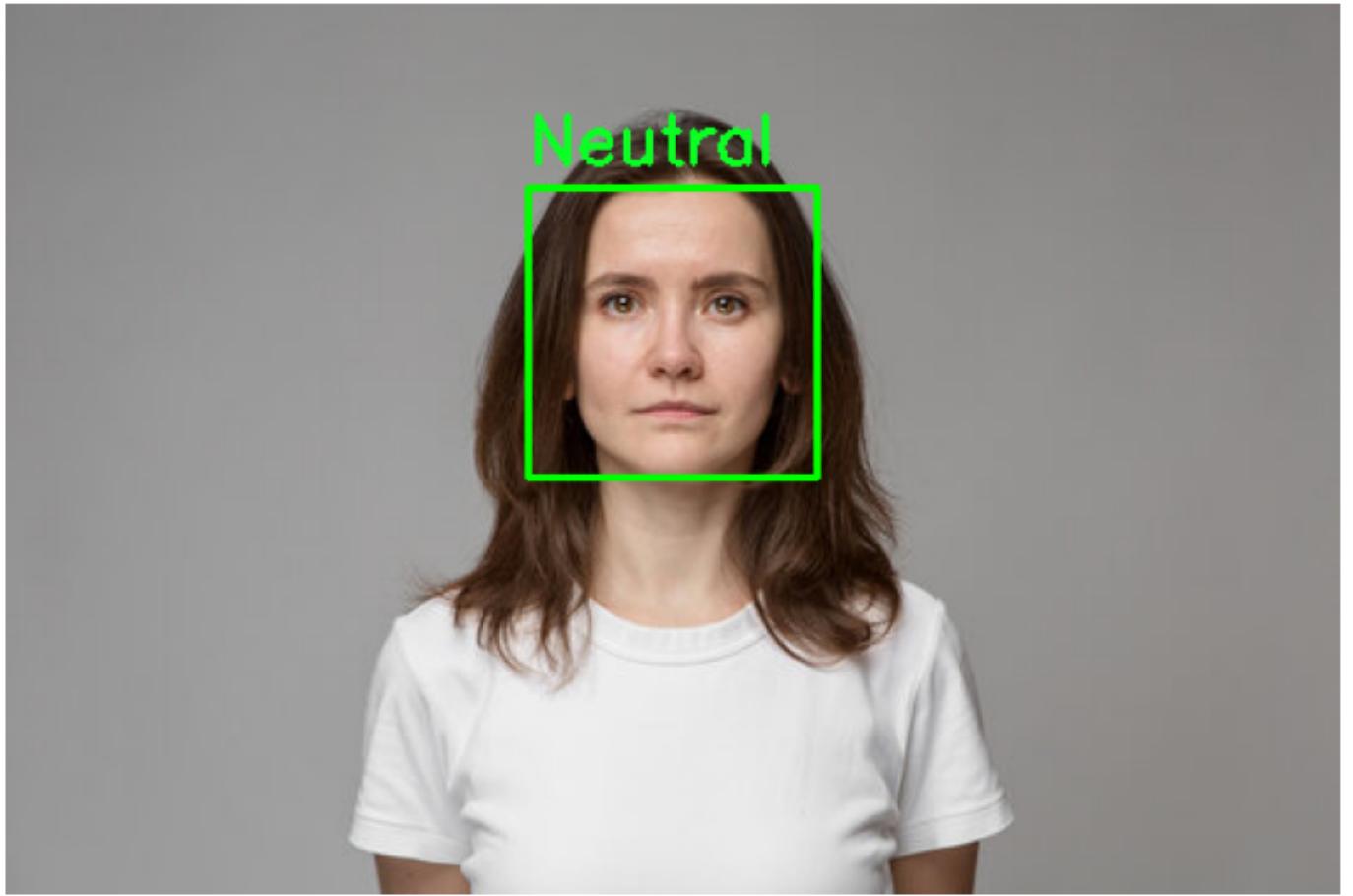


```
In [ ]: # Test "Happy" expression image
happy_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Vis
display_image_with_emotions(happy_image_path)
```

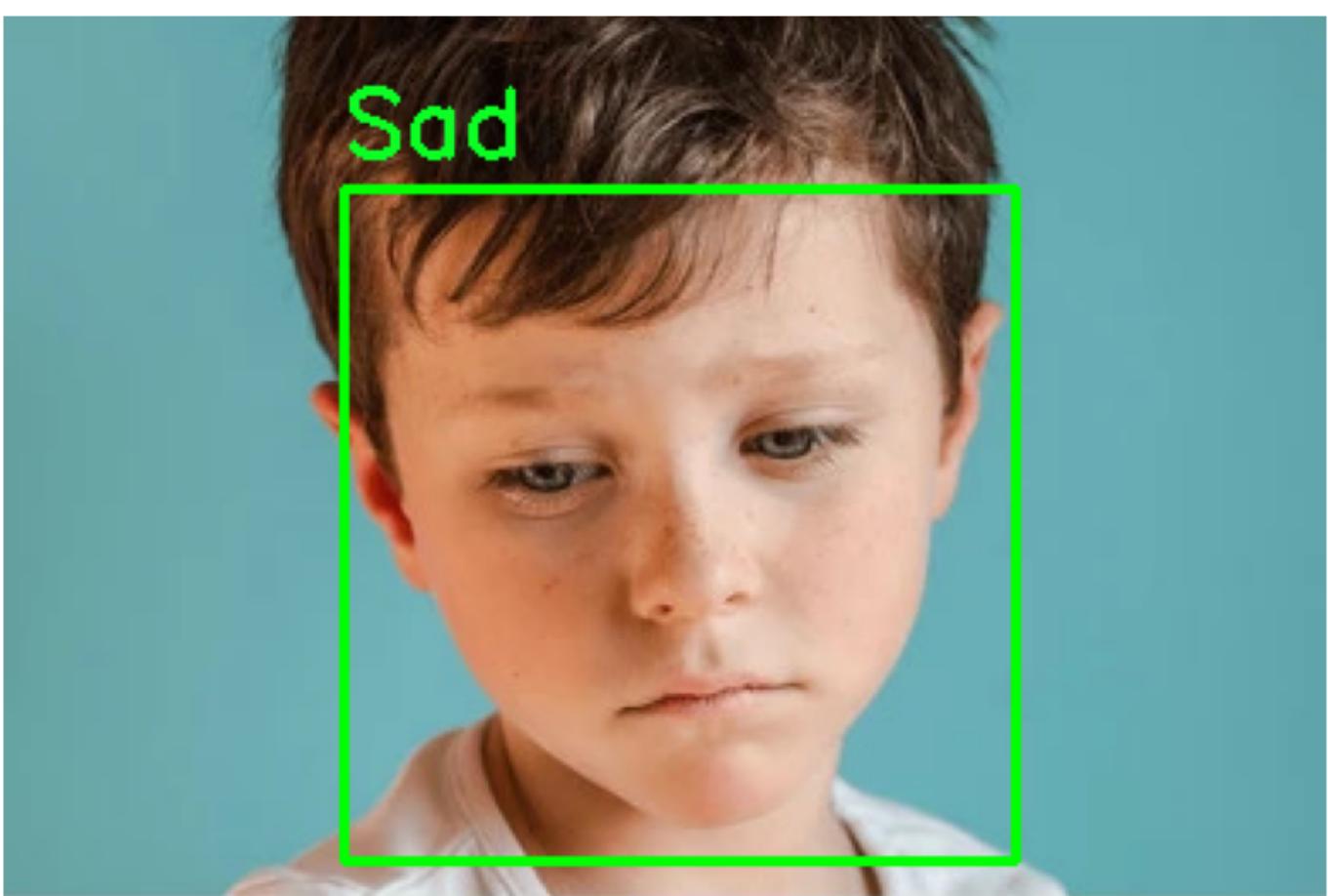
1/1 [=====] - 0s 191ms/step



```
In [ ]: # Test "Neutral" expression image  
neutral_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-V  
display_image_with_emotions(neutral_image_path)  
1/1 [=====] - 0s 298ms/step
```



```
In [ ]: # Test "Sad" expression image  
sad_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Visio  
display_image_with_emotions(sad_image_path)  
1/1 [=====] - 0s 186ms/step
```



shutterstock.com · 2145256465

```
In [ ]: # Test "Surprise" expression image
surprise_image_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-
display_image_with_emotions(surprise_image_path)

1/1 [=====] - 0s 191ms/step
```



shutterstock.com · 763024255

```
In [ ]: # Function to predict emotion (to be used in actual Taipy App)
def predict_emotion(image):
    faces = face_classifier.detectMultiScale(image, scaleFactor=1.1, minNeighbors=5, min
emotions = []
for (x, y, w, h) in faces:
    face = image[y:y + h, x:x + w]
    face = cv2.resize(face, (224, 224))
    face = face.astype("float") / 255.0
    face = img_to_array(face)
    face = np.expand_dims(face, axis=0)

    prediction = model.predict(face)[0]
    emotion = emotion_labels[np.argmax(prediction)]
    emotions.append((x, y, w, h, emotion))
return emotions
```

```
In [ ]: def analyze_image(state):
    if state.image_file:
        nparr = np.frombuffer(state.image_file, np.uint8)
        image = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
        state.image = image
        state.emotions = predict_emotion(image)
        notify(state, 'image_source', 'render_image')
```

```
In [ ]: def render_image(state):
    if state.image is not None:
        image = state.image.copy()
        for (x, y, w, h, emotion) in state.emotions:
            cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(image, emotion, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36,
_, img_encoded = cv2.imencode('.jpg', image))
```

```
    return img_encoded.tobytes()
return None
```

```
In [ ]: # Define GUI content
image_content = """
# Emotion Classifier

<|layout|columns=1 1|gap=10px|
<
<|label|Image Upload|
<|file|value=image_file|on_change=analyze_image|accept=image/*|label=Upload Image|>

|>
<
<|image|source=image_source|width=600px|>
|>
|>

"""

```

```
In [ ]: # Initialize GUI
state = {
    'image_file': None,
    'image': None,
    'emotions': []
}

gui = Gui(page=image_content)
gui.run(state=state, title="Emotion Classifier App", host="0.0.0.0", port=5000, image_so
[2024-06-29 10:54:25][Taipy][INFO] Running in 'single_client' mode in notebook environment
INFO:Taipy:Running in 'single_client' mode in notebook environment
[2024-06-29 10:54:25][Taipy][INFO] Gui server has been stopped.
INFO:Taipy:Gui server has been stopped.
[2024-06-29 10:54:25][Taipy][INFO] * Server starting on http://localhost:5000
INFO:Taipy: * Server starting on http://localhost:5000
It seems that the gevent monkey-patching is being used.
Please set an environment variable with:
GEVENT_SUPPORT=True
to enable gevent support in the debugger.
It seems that the gevent monkey-patching is being used.
Please set an environment variable with:
GEVENT_SUPPORT=True
to enable gevent support in the debugger.
It seems that the gevent monkey-patching is being used.
Please set an environment variable with:
GEVENT_SUPPORT=True
to enable gevent support in the debugger.
```

---

## 3. Deployment of Emotion Classifier Taipy App on Render Platform

### 1. Steps for Deployment on Render Platform

Here's a detailed breakdown of deploying the Emotion Classifier Taipy App on Render Platform:

## Prerequisites:

- A Render account (free tier available) - <https://render.com/>
- Emotion Classifier Taipy App Python code
- Dependencies listed in a `requirements.txt` file

## Choosing a Packaging Method:

There are two main ways to package your Taipy app for deployment on Render:

### 1. Zip Deployment (Simpler but might not be ideal for large dependencies):

- Create a zip file containing:
  - Emotion Classifier Taipy App Python code
  - The `requirements.txt` file listing all dependencies

### 2. Virtual Environment Deployment (Recommended for better dependency isolation):

- Create a virtual environment using tools like `venv` or `virtualenv`.
- Install all the dependencies (including Taipy) within this virtual environment.
- Zip the entire virtual environment directory along with Taipy App Python code.

## Creating a Web Service on Render:

1. Login to your Render account and navigate to the "**Web Services**" section.
2. Click on "New" and select "**Web Service**".
3. Choose a name for your web service (e.g., "Emotion-Classifier-App").
4. Select "**Python**" as the runtime.
5. Under "**Build**", choose "**Dockerfile**" (Render will handle building the Docker image for you).

## Deployment:

1. In the "**Deploy**" tab, select the deployment method based on your packaging choice:
  - **Zip Deployment:** Choose "**Zip**" and upload the zip file containing the app code and `requirements.txt`.
  - **Virtual Environment Deployment:** Choose "**Directory**" and upload the zipped virtual environment directory.
2. Render will automatically:
  - Build a Docker image containing the app and its dependencies.
  - Deploy the app to a containerized environment.
3. Once the deployment is successful, you'll see a green checkmark and details like allocated resources and a public URL.

## Accessing the App:

The public URL provided by Render allows you to access the deployed Taipy app online. Anyone with the URL can use the app to upload images and get emotion predictions.

## Additional Notes:

- **Environment Variables:** If the app code uses any environment variables (like API keys), you can set them in Render's environment variable settings.

- **Scaling:** Render allows you to scale the app based on traffic. You can adjust resource allocation (CPU, memory) as needed.
- **requirements.txt :** This file is crucial for both deployment methods. It lists all the Python packages your app requires to run.

## 2. Emotion Classifier Taipy App Python code

Our Emotion Classifier Taipy App Python code `run_emotion_classifier_on_images.py` is given below:

In [10]: `!cat run_emotion_classifier_on_images.py`

```
#!/usr/bin/env python3

# Emotion Classifier Taipy App - Run Emotion Classifier on user uploaded images & get predictions

# Imports
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import img_to_array
from taipy.gui import Gui, notify

# Load your pre-trained model
model = tf.keras.models.load_model("ResNet50_Transfer_Learning_40_Epochs.keras")

# Emotion labels
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']

# Initialize the face classifier
face_classifier = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

# Function to predict emotion
def predict_emotion(image):
    faces = face_classifier.detectMultiScale(image, scaleFactor=1.1, minNeighbors=5, minSize=(100, 100), flags=cv2.CASCADE_SCALE_IMAGE)
    emotions = []
    for (x, y, w, h) in faces:
        face = image[y:y + h, x:x + w]
        face = cv2.resize(face, (224, 224))
        face = face.astype("float") / 255.0
        face = img_to_array(face)
        face = np.expand_dims(face, axis=0)

        prediction = model.predict(face)[0]
        emotion = emotion_labels[np.argmax(prediction)]
        emotions.append((x, y, w, h, emotion))
    return emotions

def analyze_image(state):
    if state.image_file:
        nparr = np.frombuffer(state.image_file, np.uint8)
        image = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
        state.image = image
        state.emotions = predict_emotion(image)
        notify(state, 'image_source', 'render_image')
```

```

def render_image(state):
    if state.image is not None:
        image = state.image.copy()
        for (x, y, w, h, emotion) in state.emotions:
            cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(image, emotion, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36,
255, 12), 2)
        _, img_encoded = cv2.imencode('.jpg', image)
        return img_encoded.tobytes()
    return None

# Define GUI content
image_content = """
# Emotion Classifier

<|layout|columns=1 1|gap=10px|
<|
<|label|Image Upload|>
<|file|value=image_file|on_change=analyze_image|accept=image/*|label=Upload Image|>

|>
<|
<|image|source=image_source|width=600px|>
|>
|>

"""

# Initialize GUI
state = {
    'image_file': None,
    'image': None,
    'emotions': []
}

gui = Gui(page=image_content)
gui.run(state=state, title="Emotion Classifier App", host="0.0.0.0", port=5000, image_so
urce=render_image)

```

### 3. Requirements file

Our requirements file `requirements.txt` is given below:

In [9]: `!cat requirements.txt`

```

tensorflow
keras
numpy
matplotlib
opencv-python
taipy
taipy-gui

```

### 4. Storing Emotion Classifier Model in Git LFS

Our Emotion Classifier model (`ResNet50_Transfer_Learning_40_EPOCHS.keras`) has a size of 303 MB. The maximum individual file size for GitHub repositories is 100 MB. This means any single file larger than 100 MB cannot be uploaded directly to a GitHub repository. Hence our model cannot be uploaded directly to GitHub repo.

Hence, the model is stored in Git LFS (Large File Storage). Git LFS is an open-source extension for Git that helps manage large files within your version control system. Here's what happens when you use Git LFS for your model:

#### Stored in GitHub Repo:

- **Pointer File:** Git LFS won't store the actual model file itself in the GitHub repository. Instead, it creates a small pointer file (usually a few bytes) containing information about the model file. This includes:
  - **Git LFS pointer:** This acts as a reference or address for the actual model file location.
  - **Hash:** A unique identifier for the model file version.

#### Git LFS Storage:

- **Remote Storage:** The actual 303 MB model file is stored on a separate server managed by Git LFS or your Git hosting provider (like GitHub). This frees up space in your Git repository and keeps it manageable.

#### Key Points:

- You'll need to configure Git LFS on your local machine and in your Git repository on GitHub. Refer to the official Git LFS documentation <https://git-lfs.com/> for detailed instructions.
- Git LFS seamlessly integrates with your Git workflow. You can commit, push, and pull changes to your model file just like any other file, but Git LFS handles the storage and retrieval behind the scenes.
- When you clone the repository, Git LFS automatically downloads the model file from the remote storage based on the information in the pointer file.

By using Git LFS, you can effectively track and manage your large model file within your Git repository on GitHub without exceeding file size limitations.

## 5. Failed Render Deployment

Render Platform assumes that Emotion Classifier model is present in the GitHub repo. Since model is actually in Git LFS, deployment in Render failed.

The screenshots of failed Render Deployment are shown below:

In [2]: # Failed Render Deployment

```
from IPython import display
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deplo
```

Out[2]:

Inbox 99+ 01-PERSONAL 03-WORK All Bookmarks

Gmail Search mail Active

Deploy failed for Emotion-Classifier-Taipy-App

Render <no-reply@render.com> to me 8:48 PM (2 hours ago)



**Deploy failed for Emotion-Classifier-Taipy-App**

Your Deploy failed for Web Service [Emotion-Classifier-Taipy-App](#).  
The Deploy used [this commit](#) with message "Added deploy/requirements.txt".

In [3]: # Failed Render Deployment

```
from IPython import display
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deplo
```

Out[3]:

dashboard.render.com/web/srv-cq029oju9rs73an4fo0/deploy/de-cq029p2ju9rs73an4fs0?r=2024-06-29%4... 01-PERSONAL 03-WORK All Bookmarks

Render Dashboard Blueprints Env Groups + New ancilcleetus.work@gmail.com

WEB SERVICE Emotion-Classifier-Taipy-App Python 3 Free Upgrade your instance → Connect Manual Deploy

ancilcleetus / My-Learning-Journey main https://emotion-classifier-taipy-app.onrender.com

Events Logs Disks Environment Shell Previews Jobs Metrics

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. Upgrade now

June 29, 2024 at 8:44 PM Failed db42b86 Added deploy/requirements.txt

Exited with status 1 while running your code. Read our [docs](#) for common ways to troubleshoot your deploy.

All logs Search Jun 29, 8:43 PM - 8:49 PM GMT+5:30

Jun 29 08:44:46 PM => Cloning from https://github.com/ancilcleetus/My-Learning-Journey  
Jun 29 08:44:49 PM => Checking out commit db42b86aa316fd50c0740f6c8dd328d7bfb6a in branch main

In [4]: # Failed Render Deployment

```
from IPython import display
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deplo
```

Out[4]:

Emotion-Classifier-Taipy-App / Logs

## Logs

All logs ▼ Q Search Jun 29, 8:43 PM - 8:49 PM ▼ GMT+5:30 ↑

```
Jun 29 08:44:46 PM ⓘ INFO    ==> Cloning from https://github.com/ancileetus/My-Learning-Journey
Jun 29 08:44:49 PM ⓘ INFO    ==> Checking out commit db42b86aa316fd50c0740f6c8dd328d7bfefbd6a in branch main
Jun 29 08:44:51 PM ⓘ INFO    ==> Using Python version 3.11.9 (default)
Jun 29 08:44:51 PM ⓘ INFO    ==> Docs on specifying a Python version: https://render.com/docs/python-version
Jun 29 08:44:54 PM ⓘ INFO    ==> Using Poetry version 1.7.1 (default)
Jun 29 08:44:54 PM ⓘ INFO    ==> Docs on specifying a Poetry version: https://render.com/docs/poetry-version
Jun 29 08:44:54 PM ⓘ INFO    ==> Running build command 'pip3 install -r requirements.txt'...
Jun 29 08:44:54 PM ⓘ INFO    Collecting tensorflow (from -r requirements.txt (line 1))
Jun 29 08:44:54 PM ⓘ INFO        Downloading tensorflow-2.16.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.2 kB)
Jun 29 08:44:54 PM ⓘ INFO    Collecting keras (from -r requirements.txt (line 2))
Jun 29 08:44:54 PM ⓘ INFO        Downloading keras-3.4.1-py3-none-any.whl.metadata (5.8 kB)
Jun 29 08:44:55 PM ⓘ INFO    Collecting numpy (from -r requirements.txt (line 3))
Jun 29 08:44:55 PM ⓘ INFO        Downloading numpy-2.0.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
Jun 29 08:44:55 PM ⓘ INFO        ━━━━━━━━ 60.9/60.9 kB 3.4 MB/s eta 0:00:00
Jun 29 08:44:55 PM ⓘ INFO    Collecting matplotlib (from -r requirements.txt (line 4))
Jun 29 08:44:55 PM ⓘ INFO        Downloading matplotlib-3.9.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
Jun 29 08:44:55 PM ⓘ INFO    Collecting opencv-python (from -r requirements.txt (line 5))
Jun 29 08:44:55 PM ⓘ INFO        Downloading opencv_python-4.10.0.84-cp37abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (20 kB)
```

```
In [5]: # Failed Render Deployment
```

```
from IPython import display
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deplo
```

Out[5]:

01-PERSONAL 03-WORK All Bookmarks

Emotion-Classifier-Taipy-App / Logs

Logs

All logs Search Jun 29, 8:43 PM - 8:49 PM GMT+5:30 ↗

```
0, keras, flask-testui, flask-cors, cookiecutter, apispec-webframeworks, tensorflow, python-socketio, boto3, taipy-core, flask-socketio, taipy-rest, taipy-gui, taipy

y Jun 29 08:46:16 PM ⓘ INFO Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.3 absl-py-2.1.0 aniso8601-9.0.1 apispec-6.4.0 apispec-webframeworks-1.0.0 arrow-1.3.0 astunparse-1.6.3 attrs-23.2.0 automat-22.10.0 bidict-0.23.1 binaryornot-0.4.4 blinker-1.8.2 boto3-1.34.34 botocore-1.34.136 certifi-2024.6.2 chardet-5.2.0 charset-normalizer-3.3.2 click-8.1.7 constantly-23.10.4 contourpy-1.2.1 cookiecutter-2.5.0 cycler-0.12.1 deepdiff-6.7.1 dnspython-2.6.1 et-xmlfile-1.1.0 flask-2.0.2 flask-cors-4.0.0 flask-restful-0.3.10 flask-socketio-5.3.0 flatbuffers-24.3.25 fonttools-4.53.0 gast-0.6.0 gevent-23.9.1 gevent-websocket-0.10.1 gitignore-0.1.11 google-pasta-0.2.0 greenlet-3.0.3 grpcio-1.64.1 h11-0.14.0 h2py-3.11.0 hyperlink-21.0.0 idna-3.7 incremental-22.10.0 itsdangerous-2.2.0 jmespath-1.0.1 keras-1.2.0 kiwisolver-0.1.26.4 kthread-0.2.3 libclang-18.1.1 markdown-3.5.2 markdown-it-py-3.0.0 marshmallow-3.20.2 matplotlib-3.9.0 mdurl-0.1.2 ml-dtypes-0.3.2 nameko-0.0.8 networkx-3.2.1 numpy-1.26.4 opencv-python-4.10.0.84 openpyxl-3.1.2 opt-einsum-3.3.0 optree-0.11.0 ordered-set-4.1.0 packaging-24.1 pandas-2.2.0 passlib-1.7.4 pillow-10.3.0 protobuf-4.25.3 pyarrow-15.0.0 pygments-2.18.0 pymongo-4.6.1 pypparsing-3.1.2 python-dateutil-2.9.0.post0 python-dotenv-1.0.1 python-engineio-4.9.1 pythonslugify-8.0.4 python-socketio-5.11.3 pytz-2023.3.post1 pyyaml-6.0.1 requests-2.32.0 rich-13.7.1 s3transfer-0.10.2 simple-websocket-1.0.0 six-1.16.0 sqlalchemy-2.0.25 taipy-3.1.1 taipy-config-3.1.1 taipy-core-3.1.1 taipy-gui-3.1.3 taipy-rest-3.1.1 taipy-templates-3.1.1 tensorboard-2.16.2 tensorboard-data-server-0.7.2 tensorflow-2.16.2 tensorflow-io-gcs-filesystem-0.37.0 termcolor-2.4.0 text-unidecode-1.3 toml-0.10.2 twisted-23.10.0 types-python-dateutil-2.9.0.20240316 typing-extensions-4.12.2 tzdata-2024.1 tzlocal-5.2 urllib3-2.2.2 wheel-0.43.0 wrapt-1.16.0 wsproto-1.2.0 zope.event-5.0 zope.interface-6.4.post2

Jun 29 08:46:17 PM ⓘ INFO
Jun 29 08:46:17 PM ⓘ NOTICE [notice] A new release of pip is available: 24.0 -> 24.1.1
Jun 29 08:46:17 PM ⓘ NOTICE [notice] To update, run: pip install --upgrade pip
Jun 29 08:46:20 PM ⓘ INFO => Uploading build...
Jun 29 08:46:43 PM ⓘ INFO => Build uploaded in 13s
Jun 29 08:46:47 PM ⓘ INFO => Deploying...
Jun 29 08:46:43 PM ⓘ INFO => Build successful 🎉
Jun 29 08:48:05 PM ⓘ INFO => Using Node version 20.12.2 (default)
```

```
In [6]: # Failed Render Deployment
```

```
from IPython import display  
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deplo
```

Out[6]:

In [11]:

```
# Failed Render Deployment

from IPython import display
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deploy.png")
```

Out[11]:

The screenshot shows a browser window with the URL <https://dashboard.render.com/web/srv-cq029oqju9rs73an4fo0/deploy/dep-cq029p2ju9rs73an4fs0?r=2024-06-29%4...>. The page title is "Emotion-Classifier-Taipy-App / Logs". The logs section displays deployment logs from June 29, 2024, at 8:46 PM. The logs include messages about pip upgrades, Node.js versions, Bun versions, and TensorFlow. There are multiple error messages related to CUDA drivers and GPU support.

In [11]:

```
# Failed Render Deployment

from IPython import display
display.Image("data/images/CV_Project_01_Emotion_Classifier_Keras-01-Render-Failed-Deploy.png")
```

Out[11]:

The screenshot shows a browser window with the same URL as the previous one. The logs section now shows TensorFlow warnings about finding tensorRT, followed by many INFO and WARNING messages. These messages are related to TensorFlow's saving API, Keras models, and Python's saving API. There are also standard deployment setup messages like Node.js and Bun versions.

## 4. Emotion Classifier Gradio App for Prediction on User Images

Since Taipy App on Render platform didn't work out as expected, we will deploy using Gradio. Here are some of the key advantages of Gradio for deploying machine learning models:

#### **Ease of Use:**

- **Low Code Requirement:** Gradio requires minimal coding compared to building a full-fledged web application. You can define the user interface and connect it to your model prediction function with just a few lines of code.
  - **Intuitive Interface:** Gradio provides pre-built components for common input and output types (images, text, numbers, etc.), making it easy to create user-friendly interfaces without extensive web development knowledge.

## Rapid Prototyping and Iteration:

- **Fast Development Cycle:** Gradio allows you to quickly create a functional web interface for your model, enabling rapid prototyping and testing of your ideas.
  - **Iterative Refinement:** You can easily modify your prediction function or user interface code and see the changes reflected in the Gradio interface immediately. This facilitates iterative development and refinement of your model deployment.

## **Flexibility:**

- **Supports Various Models:** Gradio integrates with a wide range of machine learning frameworks, including TensorFlow, PyTorch, Scikit-learn, and others. This allows you to deploy various types of models without framework-specific limitations.
  - **Customization Options:** While Gradio offers pre-built components, it also allows for customization of the user interface with HTML, CSS, and JavaScript. This enables you to tailor the interface to your specific needs and branding.

Gradio integrates seamlessly with Jupyter Notebook, allowing you to directly deploy your machine learning models as interactive web applications within your notebook environment.

## Benefits of Using Gradio in Jupyter Notebook:

- **Rapid Prototyping:** Experiment with different model configurations and user interfaces directly within your notebook.
  - **Interactive Exploration:** Users can interact with your model by uploading data and seeing predictions in real-time.
  - **Easy Sharing:** Share your notebook with collaborators, allowing them to interact with the model without setting up their environments.

```
In [12]: !pip3 install gradio
```

```
Collecting gradio
  Downloading gradio-4.37.2-py3-none-any.whl (12.3 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.3/12.3 MB 36.1 MB/s eta 0:00:00
Collecting aiofiles<24.0,>=22.0 (from gradio)
  Downloading aiofiles-23.2.1-py3-none-any.whl (15 kB)
Requirement already satisfied: altair<6.0,>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (4.2.2)
Collecting fastapi (from gradio)
  Downloading fastapi-0.111.0-py3-none-any.whl (91 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.0/92.0 kB 14.8 MB/s eta 0:00:00
Collecting ffmpy (from gradio)
  Downloading ffmpy-0.3.2.tar.gz (5.5 kB)
  Preparing metadata (setup.py) ... done
Collecting gradio-client==1.0.2 (from gradio)
  Downloading gradio_client-1.0.2-py3-none-any.whl (318 kB)
```

318.2/318.2 kB 32.4 MB/s eta 0:00:00

Collecting httpx>=0.24.1 (from gradio)

  Downloading httpx-0.27.0-py3-none-any.whl (75 kB) 75.6/75.6 kB 10.7 MB/s eta 0:00:00

Requirement already satisfied: huggingface-hub>=0.19.3 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.23.4)

Requirement already satisfied: importlib-resources<7.0,>=1.3 in /usr/local/lib/python3.10/dist-packages (from gradio) (6.4.0)

Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (3.1.4)

Requirement already satisfied: markupsafe~=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.1.5)

Requirement already satisfied: matplotlib~=3.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (3.7.1)

Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (1.25.2)

Collecting orjson~=3.0 (from gradio)

  Downloading orjson-3.10.5-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (144 kB) 145.0/145.0 kB 20.1 MB/s eta 0:00:00

Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from gradio) (24.1)

Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.0.3)

Requirement already satisfied: pillow<11.0,>=8.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (9.4.0)

Requirement already satisfied: pydantic>=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.7.4)

Collecting pydub (from gradio)

  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)

Collecting python-multipart>=0.0.9 (from gradio)

  Downloading python\_multipart-0.0.9-py3-none-any.whl (22 kB)

Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (6.0.1)

Collecting ruff>=0.2.2 (from gradio)

  Downloading ruff-0.5.0-py3-none-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (10.1 MB) 10.1/10.1 kB 79.4 MB/s eta 0:00:00

Collecting semantic-version~=2.0 (from gradio)

  Downloading semantic\_version-2.10.0-py2.py3-none-any.whl (15 kB)

Collecting tomlkit==0.12.0 (from gradio)

  Downloading tomlkit-0.12.0-py3-none-any.whl (37 kB)

Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.12.3)

Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (4.12.2)

Requirement already satisfied: urllib3~=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.0.7)

Collecting uvicorn>=0.14.0 (from gradio)

  Downloading uvicorn-0.30.1-py3-none-any.whl (62 kB) 62.4/62.4 kB 9.7 MB/s eta 0:00:00

Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from gradio-client==1.0.2->gradio) (2023.6.0)

Collecting websockets<12.0,>=10.0 (from gradio-client==1.0.2->gradio)

  Downloading websockets-11.0.3-cp310-cp310-manylinux\_2\_5\_x86\_64.manylinux1\_x86\_64.manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (129 kB) 129.9/129.9 kB 18.0 MB/s eta 0:00:00

Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (0.4)

Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (4.19.2)

Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (0.12.1)

Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (3.7.1)

Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from

```
httpx>=0.24.1->gradio) (2024.6.2)
Collecting httpcore==1.* (from httpx>=0.24.1->gradio)
    Downloading httpcore-1.0.5-py3-none-any.whl (77 kB)
        █████████████████████████████████████████████████████████████████ 77.9/77.9 kB 10.6 MB/s eta 0:00:00
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (3.7)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (1.3.1)
Collecting h11<0.15,>=0.13 (from httpcore==1.*->httpx>=0.24.1->gradio)
    Downloading h11-0.14.0-py3-none-any.whl (58 kB)
        █████████████████████████████████████████████████████████████████ 58.3/58.3 kB 9.0 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (3.15.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (2.31.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (4.66.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~3.0->gradio) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib~3.0->gradio) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib~3.0->gradio) (4.53.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~3.0->gradio) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~3.0->gradio) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib~3.0->gradio) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0->gradio) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0->gradio) (2024.1)
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio) (0.7.0)
Requirement already satisfied: pydantic-core==2.18.4 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio) (2.18.4)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (13.7.1)
Collecting starlette<0.38.0,>=0.37.2 (from fastapi->gradio)
    Downloading starlette-0.37.2-py3-none-any.whl (71 kB)
        █████████████████████████████████████████████████████████████████ 71.9/71.9 kB 10.9 MB/s eta 0:00:00
Collecting fastapi-cli>=0.0.2 (from fastapi->gradio)
    Downloading fastapi_cli-0.0.4-py3-none-any.whl (9.5 kB)
Collecting ujson!=4.0.2,!=4.1.0,!=4.2.0,!=4.3.0,!=5.0.0,!=5.1.0,>=4.0.1 (from fastapi->gradio)
    Downloading ujson-5.10.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (53 kB)
        █████████████████████████████████████████████████████████████████ 53.6/53.6 kB 3.9 MB/s eta 0:00:00
Collecting email_validator>=2.0.0 (from fastapi->gradio)
    Downloading email_validator-2.2.0-py3-none-any.whl (33 kB)
Collecting dnspython>=2.0.0 (from email_validator>=2.0.0->fastapi->gradio)
    Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
        █████████████████████████████████████████████████████████████████ 307.7/307.7 kB 28.2 MB/s eta 0:00:00
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (23.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (0.18.1)
```

```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib~3.0->gradio) (1.16.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.16.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx>=0.24.1->gradio) (1.2.1)
Collecting httptools>=0.5.0 (from uvicorn>=0.14.0->gradio)
  Downloading httptools-0.6.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (341 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 341.4/341.4 kB 40.8 MB/s eta 0:00:00
Collecting python-dotenv>=0.13 (from uvicorn>=0.14.0->gradio)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Collecting uvloop!=0.15.0,!0.15.1,>=0.14.0 (from uvicorn>=0.14.0->gradio)
  Downloading uvloop-0.19.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.4 MB)
    ━━━━━━━━━━━━━━━━━━━ 3.4/3.4 kB 97.2 MB/s eta 0:00:00
Collecting watchfiles>=0.13 (from uvicorn>=0.14.0->gradio)
  Downloading watchfiles-0.22.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
    ━━━━━━━━━ 1.2/1.2 kB 61.0 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.19.3->gradio) (3.3.2)
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio) (0.1.2)
Building wheels for collected packages: ffmpy
  Building wheel for ffmpy (setup.py) ... done
  Created wheel for ffmpy: filename=ffmpy-0.3.2-py3-none-any.whl size=5584 sha256=83d312ff894be0181b11189c28f60e777149e6fcfc50b9c678357ab43e54bbc43
  Stored in directory: /root/.cache/pip/wheels/bd/65/9a/671fc6dcde07d4418df0c592f8df512b26d7a0029c2a23dd81
Successfully built ffmpy
Installing collected packages: pydub, ffmpy, websockets, uvloop, ujson, tomlkit, semantic-version, ruff, python-multipart, python-dotenv, orjson, httpools, h11, dnspython, aiofiles, watchfiles, uvicorn, starlette, httpcore, email_validator, httpx, gradio-client, fastapi-cli, fastapi, gradio
Successfully installed aiofiles-23.2.1 dnspython-2.6.1 email_validator-2.2.0 fastapi-0.11.0 fastapi-cli-0.0.4 ffmpy-0.3.2 gradio-4.37.2 gradio-client-1.0.2 h11-0.14.0 httpcore-1.0.5 httpools-0.6.1 httpx-0.27.0 orjson-3.10.5 pydub-0.25.1 python-dotenv-1.0.1 python-multipart-0.0.9 ruff-0.5.0 semantic-version-2.10.0 starlette-0.37.2 tomlkit-0.12.0 ujson-5.10.0 uvicorn-0.30.1 uvloop-0.19.0 watchfiles-0.22.0 websockets-11.0.3
```

In [13]:

```
# Imports
import numpy as np
import tensorflow as tf
import cv2
from tensorflow.keras.preprocessing.image import img_to_array
import gradio as gr
```

In [14]:

```
# Mount Google Drive
from google.colab import drive
drive.mount('/gdrive')
```

Mounted at /gdrive

In [15]:

```
# Upload the best ResNet50 model
model_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Vision/02"
model = tf.keras.models.load_model(model_path)
```

In [16]:

```
# Emotion labels
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']
```

In [17]:

```
# Initialize the face classifier
face_classifier_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer
```

```
face_classifier = cv2.CascadeClassifier(face_classifier_path)
```

```
In [18]: # Function to predict emotion (function is expecting PIL image as input)
def predict_emotion(image):
    # Convert PIL image to OpenCV format (BGR)
    opencv_image = np.array(image)[:, :, ::-1].copy() # Convert RGB to BGR

    faces = face_classifier.detectMultiScale(opencv_image, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

    # Check for detected faces
    if len(faces) == 0:
        return "No faces detected"

    # Process the first face
    (x, y, w, h) = faces[0]
    face = opencv_image[y:y + h, x:x + w]
    face = cv2.resize(face, (224, 224))
    face = face.astype("float") / 255.0
    face = img_to_array(face)
    face = np.expand_dims(face, axis=0)

    prediction = model.predict(face)[0]
    emotion = emotion_labels[np.argmax(prediction)]

    return emotion
```

```
In [19]: # Define the Gradio interface
interface = gr.Interface(
    fn=predict_emotion, # Your prediction function
    inputs=gr.Image(type="pil"), # Input for uploading an image, directly compatible with most image formats
    outputs="text", # Output as text displaying the predicted emotion
    title="Emotion Classifier",
    description="Upload an Image and Discover the Emotions Within"
)
```

```
In [20]: # Launch the Gradio interface
interface.launch()
```

Setting queue=True in a Colab notebook requires sharing enabled. Setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()  
Running on public URL: <https://d0db4d376913b3e95c.gradio.live>

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (<https://huggingface.co/spaces>)



No interface is running right now

Out[20]:

---

## 5. Deployment for Real-time Prediction on Webcam Video

We want to run Emotion Classifier on a video and save output with predictions for each frame.

```
In [ ]: # Imports
import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import img_to_array
```

```
In [ ]: # Mount Google Drive
from google.colab import drive
drive.mount('/gdrive')

Mounted at /gdrive
```

```
In [ ]: # Upload best ResNet50 model
model_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Vision/02"
model = tf.keras.models.load_model(model_path)
```

```
In [ ]: # Emotion labels
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']
```

```
In [ ]: # Initialize the face classifier
face_classifier_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer
face_classifier = cv2.CascadeClassifier(face_classifier_path)
```

```
In [ ]: def predict_emotions(video_path, output_path):
    """
        Analyzes a video, predicts emotions for each frame, and saves the output video with an

    Args:
        video_path: The path to the video file.
        output_path: The path to save the output video with annotations.
    """
    # Start capturing video from the provided path
    cap = cv2.VideoCapture(video_path)

    # Get video properties for output video creation
    fps = cap.get(cv2.CAP_PROP_FPS)
    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

    # Define video writer for output
    fourcc = cv2.VideoWriter_fourcc(*'mp4v') # Adjust codec if needed (e.g., 'XVID' for a
    out = cv2.VideoWriter(output_path, fourcc, fps, (width, height))

    # Continuous loop for processing video frames
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        # Detect faces in the frame
        faces = face_classifier.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=5, min
        # Process each face detected
        for (x, y, w, h) in faces:
            # Draw a rectangle around each detected face
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            # Extract the region of interest (ROI) i.e. the face area from the frame
            face = frame[y:y + h, x:x + w]
            # Resize the ROI to the size expected by the model (224x224 pixels in this case)
            face = cv2.resize(face, (224, 224))
            face = face.astype("float") / 255.0 # Normalize pixel values
            face = img_to_array(face)
            face = np.expand_dims(face, axis=0) # Add batch dimension

            prediction = model.predict(face)[0]
            emotion = emotion_labels[np.argmax(prediction)]
            label_position = (x, y - 10)

            cv2.putText(frame, emotion, label_position, cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 25

            # Write the annotated frame to the output video
            out.write(frame)

            # Break the loop if 'q' is pressed
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        # Release resources
        cap.release()
        out.release()
        cv2.destroyAllWindows()
```

```
In [ ]: # Prediction on video
video_path = "/gdrive/MyDrive/ancilcleetus-github/My-Learning-Journey/Computer-Vision/02
```

```
output_path = "Ancil_Cleetus_Classified_Emotions.mp4" # Adjust extension based on codec
predict_emotions(video_path, output_path)
```

```
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 25ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 37ms/step
1/1 [=====] - 0s 58ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 35ms/step
1/1 [=====] - 0s 36ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 35ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 36ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 37ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 37ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 23ms/step
```

1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 53ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step

1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 44ms/step



1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 49ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step

1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 50ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 50ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step

1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 45ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 28ms/step



1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step

1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 50ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 29ms/step

1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step

1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step

1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 47ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 52ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 48ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 59ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 50ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step

1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 30ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 33ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 32ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 38ms/step



1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 47ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 41ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 49ms/step  
1/1 [=====] - 0s 44ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 26ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step

1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 28ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 24ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 49ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 31ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 25ms/step

1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 25ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 29ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 21ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 34ms/step  
1/1 [=====] - 0s 40ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 35ms/step  
1/1 [=====] - 0s 37ms/step  
1/1 [=====] - 0s 38ms/step  
1/1 [=====] - 0s 47ms/step  
1/1 [=====] - 0s 46ms/step  
1/1 [=====] - 0s 36ms/step  
1/1 [=====] - 0s 42ms/step  
1/1 [=====] - 0s 39ms/step  
1/1 [=====] - 0s 43ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 27ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 23ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 21ms/step

```
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 22ms/step
1/1 [=====] - 0s 22ms/step
```

---

## 6. Model Retraining

### 1. Need for Model Retraining

Imagine you train a deep learning model to recognize different types of dogs in pictures. It sees thousands of images of poodles, bulldogs, chihuahuas, and so on, and learns to tell them apart. That's great! But what happens if, a year later, a new breed of dog becomes super popular?

Here's where model retraining comes in. It's like teaching your model new tricks. You feed it fresh data with pictures of this new dog breed, and the model adjusts itself to recognize them as well.

Here's why retraining is important for deep learning models:

- **The world changes:** Trends, fashions, even the way people speak can evolve. If your model's data is outdated, it won't understand these changes and its predictions might become inaccurate.
- **New data, new knowledge:** Just like us, deep learning models learn best with more information. Retraining with fresh data keeps your model up-to-date and improves its overall accuracy.

Here's a simple analogy: Imagine a child learning about animals for the first time. They see a dog and a cat, and they learn the difference. But if they only ever see those two animals, they might get confused when they encounter a horse for the first time. By showing them more animals, they can expand their knowledge and become better at identifying different creatures.

Retraining a deep learning model is similar. It's about giving it more information so it can stay relevant and make the best possible predictions.

### 2. When to do Model Retraining ?

There are two main approaches to decide when to retrain a deep learning model:

#### 1. Scheduled Retraining:

- This is like setting a regular checkup for your model. You retrain it at a predefined time interval, like every month or quarter.
- This is useful when you expect the underlying data to change steadily over time, like customer buying habits or social media trends.
- **Benefit:** Simple to implement and ensures your model is updated regularly.

- **Drawback:** Retraining too often can be computationally expensive and unnecessary if the data isn't changing significantly.

## 2. Trigger-Based Retraining:

- This is more like going to the doctor when you feel sick. You monitor the model's performance and trigger a retrain when it falls below a certain threshold.
- Performance metrics can be accuracy, error rate, or any measure specific to your task.
- This is useful when data changes can be unpredictable, or when the cost of retraining is high.
- **Benefit:** More efficient use of resources, retraining only when truly needed.
- **Drawback:** Requires setting up monitoring systems and defining clear thresholds for triggering retraining.

Here are some additional factors to consider:

- **Data Drift:** This refers to a change in the underlying data distribution that the model was trained on. For example, if your model is trained on images with good lighting and then encounters many dark photos, its performance might suffer. Data drift can trigger retraining.
- **Concept Drift:** This is a more fundamental change, where the very meaning of what the model is predicting has shifted. For instance, a model trained for spam emails might need retraining if spam tactics evolve significantly.

Ultimately, the best time to retrain your deep learning model depends on your specific application and the trade-off between keeping the model up-to-date and the cost of retraining. It's often a good idea to experiment with both scheduled and trigger-based retraining to find the optimal approach for your needs.

## 3. How to do Model Retraining ?

Retraining a deep learning model involves several steps, but here's a simplified breakdown:

1. **Gather New Data:** This is the fresh information you'll use to update your model. The data should be relevant to your task and address any potential data drift or concept drift. Ensure the new data is high-quality and properly formatted for your model.
2. **Prepare the Data:** Depending on your framework and setup, you might need to pre-process the new data. This could involve cleaning, normalization, or any other transformations you applied to the original training data.
3. **Load the Model:** Use your deep learning framework (TensorFlow, PyTorch, etc.) to load the previously trained model. This serves as the starting point for retraining.
4. **Define the Retraining Process:** Here, you specify how much of the original model to keep and how much to update with the new data. There are different approaches, but a common technique is to freeze the initial layers (which contain more general features) and retrain the later layers (which handle more specific details) on the new data.
5. **Retrain the Model:** This involves feeding the new data to the model and adjusting its internal parameters to improve its performance on the updated task. You'll likely use the same optimization algorithms you used for initial training.
6. **Evaluate the Retrained Model:** Don't just assume retraining worked! Test the retrained model on a separate dataset to see if its performance has improved on the task you care about. You might need to

adjust the retraining process or data selection if the results aren't satisfactory.

7. **Deployment:** Once you're happy with the retrained model's performance, you can deploy it back into production to make predictions on new data.

## 4. A/B Testing

A/B Testing is a controlled experiment where you compare two or more versions of something (like a website layout or a machine learning model) to see which one performs better. In the context of machine learning, you might compare a new, retrained model against the currently deployed model.

Once you've retrained your model, you can use A/B testing to compare its performance against the original model in a real-world setting. This allows you to validate the effectiveness of the retraining process before fully deploying the new model.

Consider a scenario where our deployed model serves 100,000 users. We've retrained a new model and want to rigorously validate its effectiveness before fully integrating it.

A/B testing provides a robust approach. We'll initially expose the retrained model to a controlled group of 10,000 users, while the remaining 90,000 users continue using the deployed model. Positive feedback from the initial group will prompt a measured rollout, gradually increasing the user base exposed to the retrained model (20,000, then 30,000, and so on) until all 100,000 users are transitioned.

However, if negative feedback arises at any stage, we'll strategically revert to the deployed model. This triggers a reevaluation of the retraining process, ensuring the next iteration addresses the identified shortcomings. Only once all users are satisfied with the retrained model's performance will we fully replace the deployed model.

---