```
!pip install flask-ngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from flask-ngrok) (2.27.1)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.9/dist-packages (from flask-ngrok) (2.2.3)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-ngrok)
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.8->flask-n
Requirement already satisfied: importlib-metadata>=3.6.0 in /usr/local/lib/python3.9/dist-packages (from Flask>=0.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->flask-ngrok)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->fla
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->flask-
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests-
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata>=3.6.0
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=3.0->Flask>
Installing collected packages: flask-ngrok
Successfully installed flask-ngrok-0.0.25
```

```
from io import BytesIO
from IPython.display import display
from PIL import Image
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions

import ipywidgets as widgets
import io
import matplotlib.pyplot as plt
import numpy as np
import requests
import tensorflow as tf
import tensorflow_hub as hub
import time


content_image = None # This needs to be in global scope
img_path = 'image.png'

def button_click(change):
    global content_image
    img = Image.open(io.BytesIO(uploader.data[-1]))
    content_image = img
    img.save(img_path)

uploader = widgets.FileUpload()
show_button = widgets.Button(description='Upload image')
show_button.on_click(button_click)
```

```
widgets.VBox([widgets.Label('Upload a content image (must be an RGB or RGBA image). High-res images might take more time
```

Upload a content image (must be an RGB or RGBA image). High-res images might take more time to be p…

Upload (1)

Upload image

```
import os
os.chdir('/content/drive/MyDrive/Alzheimer_s Dataset/test/MildDemented')
```

```
img = img_path
if content_image is None:
    img = "https://storage.googleapis.com/tomorrow-city/assets/migration/2019/04/architecture-buildings-cars.jpg"
```

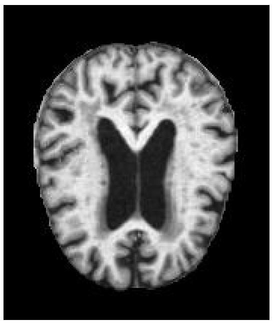
```
import os

haarcascades = os.path.join(os.path.dirname( "/content/drive/MyDrive/Alzheimer_s Dataset/test/MildDemented"))


import cv2
from PIL import ImageTk, Image
import tkinter as tk
from tkinter.filedialog import askopenfilename


def browse():
    filename = askopenfilename(initialdir="./", title="select a file",
                           filetypes=(("png files","*.png"),("allfiles","*.*")))
    if not filename:
        return  # User didn't select a file.

    tk.Label(root, text=filename).pack()
    my_image = ImageTk.PhotoImage(Image.open(filename))
    img_lbl = tk.Label(image=my_image)
    img_lbl.img = my_image  # Save reference to image.
    img_lbl.pack()

    img = cv2.imread(filename)
    show_image(img)

def show_image(img):
    cv2.imshow(" ", img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
img = img_path
if content_image is None:
    img = ("/content/drive/MyDrive/Alzheimer_s Dataset/test/MildDemented/26 (23).jpg")
load_image:any
```

```
!wget https://upload./content/drive/MyDrive/Alzheimer_s Dataset/test/MildDemented/32 (3).jpgsvg.png
```

```
/bin/bash: -c: line 0: syntax error near unexpected token `('
/bin/bash: -c: line 0: `wget https://upload./content/drive/MyDrive/Alzheimer_s Dataset/test/MildDemented/32 (3).jp
```

```
from IPython.display import Image
Image('/content/drive/MyDrive/Alzheimer_s Dataset/test/MildDemented/26 (20).jpg')
```



```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.models import load_model

#empty model
classifier = Sequential()
```

```
#add layers, start with hidden layer and first deep layer
p = 0.1
```

```
from sklearn.metrics import classification_report
```

```
import tensorflow as tf
import keras
model = keras.models.load_model
```

```
m = tf.keras.Sequential
```

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(6, activation=tf.nn.softmax)
])
```

```
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics=['accuracy'])
```

```
epochs = 50
```

```
from keras.callbacks import History
```

```
import tensorflow as tf
from tensorflow import keras
import warnings
from tensorflow.keras import layers
```

```
# Load the dataset
train_data = keras.preprocessing.image_dataset_from_directory(
    '/content/drive/MyDrive/Alzheimer_s Dataset/train',
     image_size=(150, 150),
    batch_size=32,
    shuffle=True,
    seed=42,
    validation_split=0.2,
    subset='training'
)
```

```
     Found 5121 files belonging to 4 classes.
     Using 4097 files for training.
```

```
test_data = keras.preprocessing.image_dataset_from_directory(
    '/content/drive/MyDrive/Alzheimer_s Dataset/test',
     image_size=(150, 150),
    batch_size=32,
    shuffle=True,
    seed=42,
    validation_split=0.2,
    subset='validation'
)
```

```
     Found 1279 files belonging to 4 classes.
     Using 255 files for validation.
```

```
normalization_layer = layers.experimental.preprocessing.Rescaling(1./255)
train_data = train_data.map(lambda x, y: (normalization_layer(x), y))
test_data = test_data.map(lambda x, y: (normalization_layer(x), y))
```

```
model = keras.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(150,150,3)),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(128, (3,3), activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(4, activation='softmax')
])
```

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
history = model.fit(train_data, validation_data=test_data, epochs=50)
```

```
Epoch 1/50
129/129 [==============================] - 879s 6s/step - loss: 1.0159 - accuracy: 0.5043 - val_loss: 0.9684 - v
Epoch 2/50
129/129 [==============================] - 230s 2s/step - loss: 0.8065 - accuracy: 0.6329 - val_loss: 1.0432 - v
Epoch 3/50
129/129 [==============================] - 230s 2s/step - loss: 0.5026 - accuracy: 0.7886 - val_loss: 1.1287 - v
Epoch 4/50
129/129 [==============================] - 227s 2s/step - loss: 0.3050 - accuracy: 0.8804 - val_loss: 1.7394 - v
Epoch 5/50
129/129 [==============================] - 228s 2s/step - loss: 0.1655 - accuracy: 0.9412 - val_loss: 1.3872 - v
Epoch 6/50
129/129 [==============================] - 229s 2s/step - loss: 0.0493 - accuracy: 0.9866 - val_loss: 2.1409 - v
Epoch 7/50
129/129 [==============================] - 226s 2s/step - loss: 0.0151 - accuracy: 0.9973 - val_loss: 2.1102 - v
Epoch 8/50
129/129 [==============================] - 226s 2s/step - loss: 0.0029 - accuracy: 1.0000 - val_loss: 2.8591 - v
Epoch 9/50
129/129 [==============================] - 223s 2s/step - loss: 7.4450e-04 - accuracy: 1.0000 - val_loss: 3.0377
Epoch 10/50
129/129 [==============================] - 227s 2s/step - loss: 3.7265e-04 - accuracy: 1.0000 - val_loss: 3.1281
Epoch 11/50
129/129 [==============================] - 229s 2s/step - loss: 2.7615e-04 - accuracy: 1.0000 - val_loss: 3.1754
Epoch 12/50
129/129 [==============================] - 227s 2s/step - loss: 2.1515e-04 - accuracy: 1.0000 - val_loss: 3.2536
Epoch 13/50
129/129 [==============================] - 228s 2s/step - loss: 1.7687e-04 - accuracy: 1.0000 - val_loss: 3.3119
Epoch 14/50
129/129 [==============================] - 227s 2s/step - loss: 1.4521e-04 - accuracy: 1.0000 - val_loss: 3.3864
Epoch 15/50
129/129 [==============================] - 224s 2s/step - loss: 1.2323e-04 - accuracy: 1.0000 - val_loss: 3.4911
Epoch 16/50
129/129 [==============================] - 230s 2s/step - loss: 1.0435e-04 - accuracy: 1.0000 - val_loss: 3.4790
Epoch 17/50
129/129 [==============================] - 226s 2s/step - loss: 8.9000e-05 - accuracy: 1.0000 - val_loss: 3.5681
Epoch 18/50
129/129 [==============================] - 224s 2s/step - loss: 7.8436e-05 - accuracy: 1.0000 - val_loss: 3.6079
Epoch 19/50
129/129 [==============================] - 226s 2s/step - loss: 6.7614e-05 - accuracy: 1.0000 - val_loss: 3.6708
Epoch 20/50
129/129 [==============================] - 226s 2s/step - loss: 5.9450e-05 - accuracy: 1.0000 - val_loss: 3.7299
Epoch 21/50
129/129 [==============================] - 225s 2s/step - loss: 5.2417e-05 - accuracy: 1.0000 - val_loss: 3.7282
Epoch 22/50
129/129 [==============================] - 229s 2s/step - loss: 4.6663e-05 - accuracy: 1.0000 - val_loss: 3.7879
Epoch 23/50
129/129 [==============================] - 229s 2s/step - loss: 4.2430e-05 - accuracy: 1.0000 - val_loss: 3.8283
Epoch 24/50
129/129 [==============================] - 224s 2s/step - loss: 3.6940e-05 - accuracy: 1.0000 - val_loss: 3.8611
Epoch 25/50
129/129 [==============================] - 223s 2s/step - loss: 3.2764e-05 - accuracy: 1.0000 - val_loss: 3.9148
Epoch 26/50
129/129 [==============================] - 227s 2s/step - loss: 2.9399e-05 - accuracy: 1.0000 - val_loss: 3.9813
Epoch 27/50
129/129 [==============================] - 224s 2s/step - loss: 2.6265e-05 - accuracy: 1.0000 - val_loss: 4.0474
Epoch 28/50
129/129 [==============================] - 223s 2s/step - loss: 2.5231e-05 - accuracy: 1.0000 - val_loss: 4.0617
Epoch 29/50
```

```
Executing (2h 29m 37s) <cell line: 1> › error_handler() › fit() › error_handler() › __call__() › _call() › __call__() › _call_flat() › call() › quick_execute() ...
```