import tensorflow as tf

from tensorflow import keras

import numpy as np

# Load MNIST dataset

(x\_train, y\_train), (x\_test, y\_test) = keras.datasets.mnist.load\_data()

# Preprocess data

x\_train = x\_train.reshape((-1, 28 \* 28))

x\_train = x\_train.astype('float32') / 255

x\_test = x\_test.reshape((-1, 28 \* 28))

x\_test = x\_test.astype('float32') / 255

y\_train = keras.utils.to\_categorical(y\_train)

y\_test = keras.utils.to\_categorical(y\_test)

# Define model architecture

model = keras.Sequential([

    keras.layers.Dense(512, activation='relu', input\_shape=(28 \* 28,)),

    keras.layers.Dropout(0.2),

    keras.layers.Dense(10, activation='softmax')

])

# Compile model

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Train model

model.fit(x\_train, y\_train, epochs=50, batch\_size=128, validation\_data=(x\_test, y\_test))

# Save model

# model.save('my\_model.h5')

from tensorflow import keras

import numpy as np

from PIL import Image

from google.colab import files

# Upload image

uploaded\_file = files.upload()

# Load image

image = Image.open(list(uploaded\_file.keys())[0])

# Load trained model

#model = keras.models.load\_model('my\_model.h5')

# Convert to grayscale and resize

image = image.convert('L').resize((28, 28))

# Convert to numpy array

image\_data = np.array(image)

# Preprocess image data

image\_data = image\_data.reshape((-1, 28 \* 28))

image\_data = image\_data.astype('float32') / 255

# Make prediction

prediction = model.predict(image\_data)[0]

# Get & Print predicted class label

print("Uploaded image matches to", np.argmax(prediction))