

```

!pip install onnx opencv-python scikit-learn
!pip install tf2onnx
%matplotlib inline
from google.colab import files
import os

# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dense, Flatten, Dropout
# Helper libraries
import numpy as np
import matplotlib.pyplot as plt
import cv2
import pandas as pd

# Sklearn
from sklearn.model_selection import train_test_split # Helps with organizing data for training
from sklearn.metrics import confusion_matrix # Helps present results as a confusion-matrix
import pickle
import onnx

print(tf.__version__)

Requirement already satisfied: onnx in /usr/local/lib/python3.11/dist-packages (1.17.0)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.11/dist-packages (from onnx) (1.26.4)
Requirement already satisfied: protobuf>=3.20.2 in /usr/local/lib/python3.11/dist-packages (from onnx) (3.20.3)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: tf2onnx in /usr/local/lib/python3.11/dist-packages (1.16.1)
Requirement already satisfied: numpy>=1.14.1 in /usr/local/lib/python3.11/dist-packages (from tf2onnx) (1.26.4)
Requirement already satisfied: onnx>=1.4.1 in /usr/local/lib/python3.11/dist-packages (from tf2onnx) (1.17.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from tf2onnx) (2.32.3)
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-packages (from tf2onnx) (1.17.0)
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.11/dist-packages (from tf2onnx) (25.2.10)
Requirement already satisfied: protobuf~=3.20 in /usr/local/lib/python3.11/dist-packages (from tf2onnx) (3.20.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->tf2onnx) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->tf2onnx) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->tf2onnx) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->tf2onnx) (2025.1.31)
2.18.0

with open('X_test.pkl', 'rb') as file:
    X_test = pickle.load(file)

with open('y_test.pkl', 'rb') as file:
    y_test = pickle.load(file)

with open('X_train.pkl', 'rb') as file:
    X_train = pickle.load(file)

with open('y_train.pkl', 'rb') as file:
    y_train = pickle.load(file)

model = Sequential()
model.add(Conv2D(32, (5, 5), activation='relu', input_shape=(96, 96, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(6, activation='softmax'))

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

# Configures the model for training
model.compile(optimizer='adam', # Optimization routine, which tells the computer how to adjust the parameter values to minimize the loss
              loss='sparse_categorical_crossentropy', # Loss function, which tells us how bad our predictions are.
              metrics=['accuracy']) # List of metrics to be evaluated by the model during training and testing.

```

```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 92, 92, 32)	832
max_pooling2d_3 (MaxPooling2D)	(None, 46, 46, 32)	0
dropout_2 (Dropout)	(None, 46, 46, 32)	0
conv2d_4 (Conv2D)	(None, 44, 44, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 22, 22, 64)	0
dropout_3 (Dropout)	(None, 22, 22, 64)	0
conv2d_5 (Conv2D)	(None, 20, 20, 64)	36,928
max_pooling2d_5 (MaxPooling2D)	(None, 10, 10, 64)	0
flatten_1 (Flatten)	(None, 6400)	0
dense_2 (Dense)	(None, 64)	409,664
dense_3 (Dense)	(None, 6)	390

Total params: 466,310 (1.78 MB)

Trainable params: 466,310 (1.78 MB)

Trains the model for a given number of epochs (iterations on a dataset) and validates it.

```
history=model.fit(X_train, y_train, epochs=5, batch_size=64, verbose=1, validation_data=(X_test, y_test))
```

```
Epoch 1/5
132/132 ————— 152s 1s/step - accuracy: 0.4640 - loss: 7.5001 - val_accuracy: 0.9944 - val_loss: 0.0246
Epoch 2/5
132/132 ————— 186s 988ms/step - accuracy: 0.9862 - loss: 0.0454 - val_accuracy: 0.9972 - val_loss: 0.0081
Epoch 3/5
132/132 ————— 137s 1s/step - accuracy: 0.9974 - loss: 0.0113 - val_accuracy: 0.9992 - val_loss: 0.0040
Epoch 4/5
132/132 ————— 144s 1s/step - accuracy: 0.9965 - loss: 0.0114 - val_accuracy: 0.9996 - val_loss: 0.0012
Epoch 5/5
132/132 ————— 139s 1s/step - accuracy: 1.0000 - loss: 4.9792e-04 - val_accuracy: 0.9992 - val_loss: 9.9292e-04
```

```
model.save('handrecognition_model.h5')
```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is deprecated.

```
model = tf.keras.models.load_model("/content/handrecognition_model.h5")
```

```
tf.saved_model.save(model, "tmp_model")
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you call `model.compile`.

```
!python -m tf2onnx.convert --saved-model tmp_model --output "handrecognition_model.onnx"
```

```
2025-02-24 21:49:04.613676: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register cuFFT factory: Attempt
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1740433744.641328 12997 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin
E0000 00:00:1740433744.649156 12997 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin
<Frozen runpy>:128: RuntimeWarning: 'tf2onnx.convert' found in sys.modules after import of package 'tf2onnx', but prior to execution
2025-02-24 21:49:09,846 - WARNING - ***IMPORTANT*** Installed protobuf is not cpp accelerated. Conversion will be extremely slow. See
2025-02-24 21:49:09.847279: E external/local_xla/xla/stream_executor/cuda/cuda_driver.cc:152] failed call to cuInit: INTERNAL: CUDA
2025-02-24 21:49:10,168 - INFO - Signatures found in model: [serving_default].
2025-02-24 21:49:10,168 - WARNING - '--signature_def' not specified, using first signature: serving_default
2025-02-24 21:49:10,168 - INFO - Output names: ['output_0']
I0000 00:00:1740433750.203375 12997 devices.cc:67] Number of eligible GPUs (core count >= 8, compute capability >= 0.0): 0
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1740433750.203677 12997 single_machine.cc:361] Starting new session
I0000 00:00:1740433750.550200 12997 devices.cc:67] Number of eligible GPUs (core count >= 8, compute capability >= 0.0): 0
I0000 00:00:1740433750.550387 12997 single_machine.cc:361] Starting new session
2025-02-24 21:49:10,617 - INFO - Using tensorflow=2.18.0, onnx=1.17.0, tf2onnx=1.16.1/15c810
2025-02-24 21:49:10,617 - INFO - Using opset <onnx, 15>
2025-02-24 21:49:10,651 - INFO - Computed 0 values for constant folding
2025-02-24 21:49:10,715 - INFO - Optimizing ONNX model
2025-02-24 21:49:10,873 - INFO - After optimization: Cast -1 (1->0), Const +1 (11->12), Identity -2 (2->0), Reshape +1 (1->2), Trans
2025-02-24 21:49:10,885 - INFO -
2025-02-24 21:49:10,885 - INFO - Successfully converted TensorFlow model tmp_model to ONNX
2025-02-24 21:49:10,885 - INFO - Model inputs: ['inputs']
2025-02-24 21:49:10,885 - INFO - Model outputs: ['output_0']
2025-02-24 21:49:10,886 - INFO - ONNX model is saved at handrecognition_model.onnx
```

```
!zip -r /content/tmp_model.zip /content/tmp_model
```



```
updating: content/tmp_model/ (stored 0%)  
updating: content/tmp_model/assets/ (stored 0%)  
updating: content/tmp_model/fingerprint.pb (stored 0%)  
updating: content/tmp_model/variables/ (stored 0%)  
updating: content/tmp_model/variables/variables.data-00000-of-00001 (deflated 8%)  
updating: content/tmp_model/variables/variables.index (deflated 54%)  
updating: content/tmp_model/saved_model.pb (deflated 85%)
```

```
from google.colab import files  
files.download("/content/handrecognition_model.h5")  
files.download("/content/handrecognition_model.onnx")  
files.download("/content/tmp_model.zip")
```

