import csv  
import numpy as np  
import tensorflow as tf  
from sklearn.model\_selection import train\_test\_split  
RANDOM\_SEED = 42  
import pandas as pd

# Specify each path

dataset = 'model/keypoint\_classifier/keypoint.csv'  
model\_save\_path = 'model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras'  
tflite\_save\_path = 'model/keypoint\_classifier/keypoint\_classifier.tflite'

# Set number of classes

NUM\_CLASSES = 10

# Dataset reading

X\_dataset = np.loadtxt(dataset, delimiter=',', dtype='float32', usecols=list(range(1, (21 \* 2) + 1)))  
# data\_set = pd.read\_csv(dataset)  
# data\_set.info()

y\_dataset = np.loadtxt(dataset, delimiter=',', dtype='int32', usecols=(0))

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_dataset, y\_dataset, train\_size=0.75)

# Model building

model = tf.keras.models.Sequential([  
 tf.keras.layers.Input((21 \* 2, )),  
 tf.keras.layers.Dropout(0.2),  
 tf.keras.layers.Dense(20, activation='relu'),  
 tf.keras.layers.Dropout(0.4),  
 tf.keras.layers.Dense(10, activation='relu'),  
 tf.keras.layers.Dense(NUM\_CLASSES, activation='softmax')  
])

model.summary() # tf.keras.utils.plot\_model(model, show\_shapes=True)

Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓  
┃ Layer (type) ┃ Output Shape ┃ Param # ┃  
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩  
│ dropout (Dropout) │ (None, 42) │ 0 │  
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤  
│ dense (Dense) │ (None, 20) │ 860 │  
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤  
│ dropout\_1 (Dropout) │ (None, 20) │ 0 │  
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤  
│ dense\_1 (Dense) │ (None, 10) │ 210 │  
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤  
│ dense\_2 (Dense) │ (None, 10) │ 110 │  
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘

Total params: 1,180 (4.61 KB)

Trainable params: 1,180 (4.61 KB)

Non-trainable params: 0 (0.00 B)

# Model checkpoint callback  
cp\_callback = tf.keras.callbacks.ModelCheckpoint(  
 model\_save\_path, verbose=1, save\_weights\_only=False)  
# Callback for early stopping  
es\_callback = tf.keras.callbacks.EarlyStopping(patience=20, verbose=1)

# Model compilation  
model.compile(  
 optimizer='adam',  
 loss='sparse\_categorical\_crossentropy',  
 metrics=['accuracy']  
)

# Model training

history = model.fit(  
 X\_train,  
 y\_train,  
 epochs=1000,  
 batch\_size=128,  
 validation\_data=(X\_test, y\_test),  
 callbacks=[cp\_callback, es\_callback]  
)

Epoch 1/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 20s 1s/step - accuracy: 0.8047 - loss: 0.5322  
Epoch 1: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.7864 - loss: 0.5836 - val\_accuracy: 0.9002 - val\_loss: 0.2437  
Epoch 2/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - accuracy: 0.8281 - loss: 0.5072  
Epoch 2: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8061 - loss: 0.5464 - val\_accuracy: 0.9204 - val\_loss: 0.2267  
Epoch 3/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 0.7656 - loss: 0.5811  
Epoch 3: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7867 - loss: 0.5578 - val\_accuracy: 0.8979 - val\_loss: 0.2419  
Epoch 4/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - accuracy: 0.7891 - loss: 0.5623  
Epoch 4: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7834 - loss: 0.5588 - val\_accuracy: 0.9097 - val\_loss: 0.2404  
Epoch 5/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.8047 - loss: 0.4710  
Epoch 5: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.7935 - loss: 0.5399 - val\_accuracy: 0.9097 - val\_loss: 0.2372  
Epoch 6/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 0.7656 - loss: 0.6289  
Epoch 6: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7593 - loss: 0.6103 - val\_accuracy: 0.9169 - val\_loss: 0.2432  
Epoch 7/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.7812 - loss: 0.5542  
Epoch 7: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7804 - loss: 0.5855 - val\_accuracy: 0.8990 - val\_loss: 0.2410  
Epoch 8/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7969 - loss: 0.6107  
Epoch 8: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8076 - loss: 0.5338 - val\_accuracy: 0.9109 - val\_loss: 0.2335  
Epoch 9/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.7266 - loss: 0.6258  
Epoch 9: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7689 - loss: 0.5674 - val\_accuracy: 0.9133 - val\_loss: 0.2322  
Epoch 10/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - accuracy: 0.7031 - loss: 0.7776  
Epoch 10: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7611 - loss: 0.6160 - val\_accuracy: 0.8872 - val\_loss: 0.2470  
Epoch 11/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - accuracy: 0.8438 - loss: 0.4573  
Epoch 11: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8074 - loss: 0.5249 - val\_accuracy: 0.9252 - val\_loss: 0.2303  
Epoch 12/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - accuracy: 0.8047 - loss: 0.5528  
Epoch 12: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7847 - loss: 0.5575 - val\_accuracy: 0.8789 - val\_loss: 0.2474  
Epoch 13/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.8125 - loss: 0.4858  
Epoch 13: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7889 - loss: 0.5343 - val\_accuracy: 0.9097 - val\_loss: 0.2323  
Epoch 14/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.7656 - loss: 0.6356  
Epoch 14: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7821 - loss: 0.5705 - val\_accuracy: 0.8895 - val\_loss: 0.2466  
Epoch 15/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 0.7422 - loss: 0.6970  
Epoch 15: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7719 - loss: 0.5818 - val\_accuracy: 0.9323 - val\_loss: 0.2293  
Epoch 16/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 1s 86ms/step - accuracy: 0.8125 - loss: 0.5431  
Epoch 16: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.7972 - loss: 0.5277 - val\_accuracy: 0.8741 - val\_loss: 0.2493  
Epoch 17/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - accuracy: 0.7734 - loss: 0.5276  
Epoch 17: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7768 - loss: 0.5669 - val\_accuracy: 0.9014 - val\_loss: 0.2430  
Epoch 18/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7188 - loss: 0.7366  
Epoch 18: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7732 - loss: 0.6107 - val\_accuracy: 0.8872 - val\_loss: 0.2437  
Epoch 19/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.8047 - loss: 0.5000  
Epoch 19: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7962 - loss: 0.5202 - val\_accuracy: 0.9276 - val\_loss: 0.2267  
Epoch 20/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 0.8047 - loss: 0.4858  
Epoch 20: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7908 - loss: 0.5305 - val\_accuracy: 0.8943 - val\_loss: 0.2399  
Epoch 21/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7734 - loss: 0.5386  
Epoch 21: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7723 - loss: 0.5592 - val\_accuracy: 0.9145 - val\_loss: 0.2355  
Epoch 22/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 0.7969 - loss: 0.5966  
Epoch 22: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7827 - loss: 0.5874 - val\_accuracy: 0.8836 - val\_loss: 0.2471  
Epoch 23/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.7891 - loss: 0.5506  
Epoch 23: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7779 - loss: 0.5764 - val\_accuracy: 0.9002 - val\_loss: 0.2442  
Epoch 24/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 0.7266 - loss: 0.6214  
Epoch 24: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7874 - loss: 0.5526 - val\_accuracy: 0.9299 - val\_loss: 0.2282  
Epoch 25/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 0.7969 - loss: 0.5423  
Epoch 25: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7868 - loss: 0.5574 - val\_accuracy: 0.9074 - val\_loss: 0.2362  
Epoch 26/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 0.7969 - loss: 0.5782  
Epoch 26: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8009 - loss: 0.5296 - val\_accuracy: 0.9050 - val\_loss: 0.2281  
Epoch 27/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 0.7188 - loss: 0.7013  
Epoch 27: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7672 - loss: 0.5995 - val\_accuracy: 0.8907 - val\_loss: 0.2389  
Epoch 28/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - accuracy: 0.7422 - loss: 0.5507  
Epoch 28: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7790 - loss: 0.5765 - val\_accuracy: 0.9133 - val\_loss: 0.2403  
Epoch 29/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.6953 - loss: 0.6569  
Epoch 29: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7729 - loss: 0.5663 - val\_accuracy: 0.9145 - val\_loss: 0.2417  
Epoch 30/1000  
 8/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.8051 - loss: 0.5813   
Epoch 30: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7922 - loss: 0.5763 - val\_accuracy: 0.9086 - val\_loss: 0.2432  
Epoch 31/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7344 - loss: 0.6241  
Epoch 31: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7882 - loss: 0.5527 - val\_accuracy: 0.9050 - val\_loss: 0.2424  
Epoch 32/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.8438 - loss: 0.5249  
Epoch 32: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7925 - loss: 0.5455 - val\_accuracy: 0.8979 - val\_loss: 0.2385  
Epoch 33/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - accuracy: 0.8047 - loss: 0.5192  
Epoch 33: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7958 - loss: 0.5514 - val\_accuracy: 0.8895 - val\_loss: 0.2371  
Epoch 34/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.7891 - loss: 0.6003  
Epoch 34: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7876 - loss: 0.5641 - val\_accuracy: 0.9097 - val\_loss: 0.2333  
Epoch 35/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 0.8203 - loss: 0.4756  
Epoch 35: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7940 - loss: 0.5239 - val\_accuracy: 0.9109 - val\_loss: 0.2370  
Epoch 36/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 1s 68ms/step - accuracy: 0.7734 - loss: 0.6409  
Epoch 36: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8036 - loss: 0.5336 - val\_accuracy: 0.8895 - val\_loss: 0.2375  
Epoch 37/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8047 - loss: 0.5226  
Epoch 37: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.7801 - loss: 0.5617 - val\_accuracy: 0.9062 - val\_loss: 0.2308  
Epoch 38/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8125 - loss: 0.5303  
Epoch 38: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7934 - loss: 0.5548 - val\_accuracy: 0.8955 - val\_loss: 0.2386  
Epoch 39/1000  
 1/20 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.7578 - loss: 0.5897  
Epoch 39: saving model to model/keypoint\_classifier/keypoint\_classifier\_10\_signs.keras  
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7974 - loss: 0.5382 - val\_accuracy: 0.9121 - val\_loss: 0.2285  
Epoch 39: early stopping

# Model evaluation  
val\_loss, val\_acc = model.evaluate(X\_test, y\_test, batch\_size=128)

7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9091 - loss: 0.2269

history.history["accuracy"]

[0.779405951499939,  
 0.7992079257965088,  
 0.7869306802749634,  
 0.7790098786354065,  
 0.785346508026123,  
 0.7671287059783936,  
 0.7885148525238037,  
 0.8043564558029175,  
 0.7750495076179504,  
 0.7762376070022583,  
 0.7920792102813721,  
 0.7865346670150757,  
 0.790099024772644,  
 0.7849504947662354,  
 0.7893069386482239,  
 0.7841584086418152,  
 0.7857425808906555,  
 0.7782177925109863,  
 0.7920792102813721,  
 0.7960395812988281,  
 0.7671287059783936,  
 0.7786138653755188,  
 0.7801980376243591,  
 0.7956435680389404,  
 0.7912871241569519,  
 0.7980198264122009,  
 0.7659406065940857,  
 0.7801980376243591,  
 0.7849504947662354,  
 0.7841584086418152,  
 0.7936633825302124,  
 0.7936633825302124,  
 0.7857425808906555,  
 0.7877227663993835,  
 0.7940593957901001,  
 0.8063366413116455,  
 0.7790098786354065,  
 0.790099024772644,  
 0.7960395812988281]

# Loading the saved model  
model = tf.keras.models.load\_model(model\_save\_path)

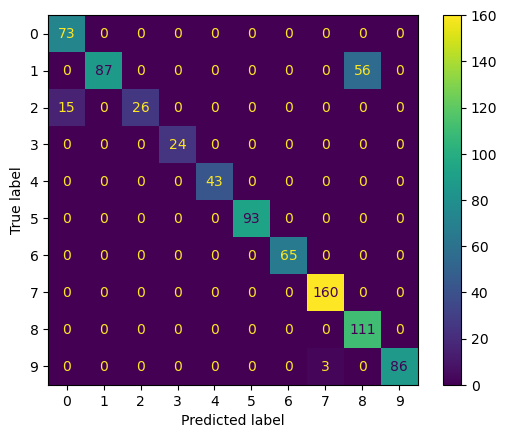
# Inference test  
predict\_result = model.predict(np.array([X\_test[0]]))  
print(np.squeeze(predict\_result))  
print(np.argmax(np.squeeze(predict\_result)),y\_test[0])

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step  
[3.4637630e-02 1.1273278e-05 2.9341480e-02 9.2777246e-01 1.4290292e-09  
 2.9864449e-03 2.3668587e-09 4.0829326e-03 1.7991461e-04 9.8782894e-04]  
3 3

# Confusion matrix

import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.metrics import confusion\_matrix, classification\_report, ConfusionMatrixDisplay  
  
def print\_confusion\_matrix(y\_true, y\_pred, report=True):  
 labels = sorted(list(set(y\_true)))  
 cmx\_data = confusion\_matrix(y\_true, y\_pred, labels=labels)  
 cm\_display = ConfusionMatrixDisplay(confusion\_matrix = cmx\_data, display\_labels = labels)  
 cm\_display.plot()  
   
 if report:  
 print('Classification Report')  
 print(classification\_report(y\_test, y\_pred))  
  
Y\_pred = model.predict(X\_test)  
y\_pred = np.argmax(Y\_pred, axis=1)  
print\_confusion\_matrix(y\_test, y\_pred)

27/27 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step   
Classification Report  
 precision recall f1-score support  
  
 0 0.83 1.00 0.91 73  
 1 1.00 0.61 0.76 143  
 2 1.00 0.63 0.78 41  
 3 1.00 1.00 1.00 24  
 4 1.00 1.00 1.00 43  
 5 1.00 1.00 1.00 93  
 6 1.00 1.00 1.00 65  
 7 0.98 1.00 0.99 160  
 8 0.66 1.00 0.80 111  
 9 1.00 0.97 0.98 89  
  
 accuracy 0.91 842  
 macro avg 0.95 0.92 0.92 842  
weighted avg 0.94 0.91 0.91 842



# Convert to model for Tensorflow-Lite

# Save as a model dedicated to inference  
model.save(model\_save\_path, include\_optimizer=False)

# Transform model (quantization)  
converter = tf.lite.TFLiteConverter.from\_keras\_model(model)  
converter.optimizations = [tf.lite.Optimize.DEFAULT]  
tflite\_quantized\_model = converter.convert()  
open(tflite\_save\_path, 'wb').write(tflite\_quantized\_model)

INFO:tensorflow:Assets written to: C:\Users\nasan\AppData\Local\Temp\tmpuhjmznmi\assets

INFO:tensorflow:Assets written to: C:\Users\nasan\AppData\Local\Temp\tmpuhjmznmi\assets

Saved artifact at 'C:\Users\nasan\AppData\Local\Temp\tmpuhjmznmi'. The following endpoints are available:  
  
\* Endpoint 'serve'  
 args\_0 (POSITIONAL\_ONLY): TensorSpec(shape=(None, 42), dtype=tf.float32, name='input\_layer')  
Output Type:  
 TensorSpec(shape=(None, 10), dtype=tf.float32, name=None)  
Captures:  
 2247251413328: TensorSpec(shape=(), dtype=tf.resource, name=None)  
 2247251415248: TensorSpec(shape=(), dtype=tf.resource, name=None)  
 2247251412560: TensorSpec(shape=(), dtype=tf.resource, name=None)  
 2247251414096: TensorSpec(shape=(), dtype=tf.resource, name=None)  
 2247251415440: TensorSpec(shape=(), dtype=tf.resource, name=None)  
 2247251412368: TensorSpec(shape=(), dtype=tf.resource, name=None)

6800

# Inference test

interpreter = tf.lite.Interpreter(model\_path=tflite\_save\_path)  
interpreter.allocate\_tensors()

# Get I / O tensor  
input\_details = interpreter.get\_input\_details()  
output\_details = interpreter.get\_output\_details()

interpreter.set\_tensor(input\_details[0]['index'], np.array([X\_test[0]]))

%%time  
# Inference implementation  
interpreter.invoke()  
tflite\_results = interpreter.get\_tensor(output\_details[0]['index'])

CPU times: total: 0 ns  
Wall time: 2.03 ms

print(np.squeeze(tflite\_results))  
print(np.argmax(np.squeeze(tflite\_results)))

[3.4637641e-02 1.1273279e-05 2.9341470e-02 9.2777258e-01 1.4290319e-09  
 2.9864409e-03 2.3668585e-09 4.0829284e-03 1.7991480e-04 9.8782859e-04]  
3