In [1]:
```
import os
import time
import numpy as np
from matplotlib import pyplot as plt
import cv2 as cv
import mediapipe as mp
import seaborn as sns
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras import models
from tensorflow.keras import layers
from sklearn.metrics import confusion_matrix ,classification_report
```

In [2]:
```
tf.__version__
```

Out[2]:
```
'2.1.0'
```

In [4]:
```
tf.test.is_gpu_available('gpu')
```

Out[4]:
```
True
```

In [5]:
```
input_types = ['palm', 'fist', 'thumbsup', 'gun', 'call']
```

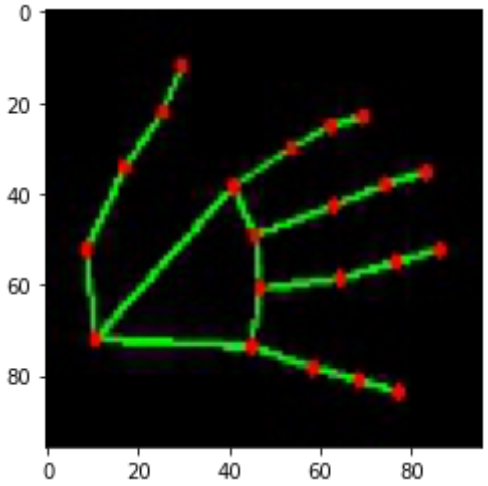
In [6]:
```
path = 'Inputs/'
data = []

for types in input_types:
    temp = path + types + '/'
    l = []
    for file in os.listdir(temp):
        if file.endswith('.jpg'):
            image_matrix = plt.imread(temp + file)
            l.append(image_matrix)
    data.append(l)
data = np.array(data)
```

In [7]:
```
data.shape
```

Out[7]:
```
(5, 2000, 96, 96, 3)
```

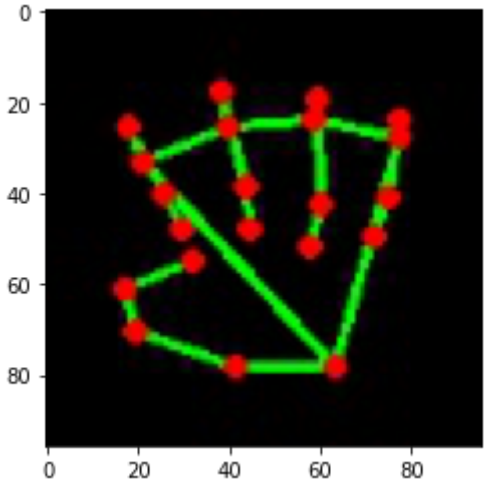
In [8]:
```
plt.imshow(data[0][30])
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x1941b3c48c8>
```



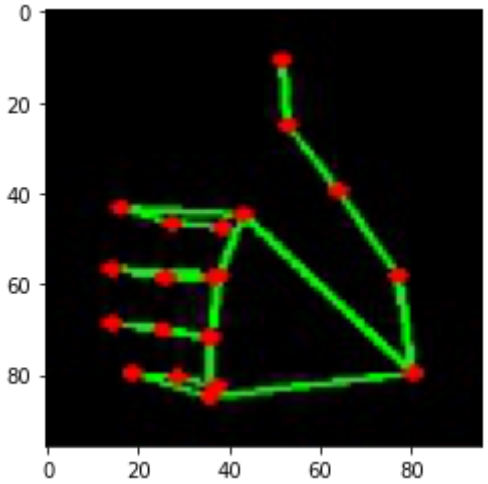
In [9]:
```
plt.imshow(data[1][70])
```

Out[9]: <matplotlib.image.AxesImage at 0x1941056ebc8>



In [10]:
```
plt.imshow(data[2][700])
```

Out[10]: <matplotlib.image.AxesImage at 0x1942be5c5c8>



In [11]:
```
data.shape
```

Out[11]:
```
(5, 2000, 96, 96, 3)
```

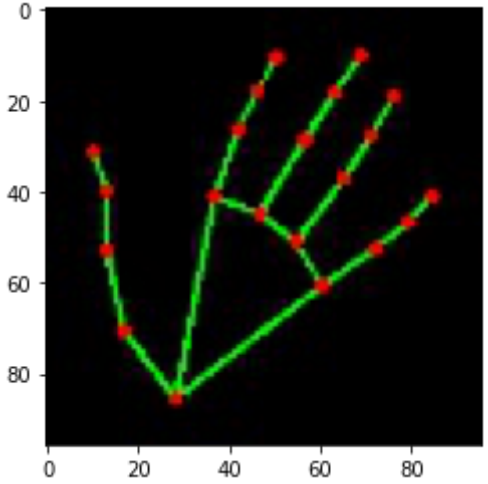
In [12]:
```
X = []
Y = []

i = 0
for class_ in data:
    for image in class_:
        X.append(image)
        Y.append(i)
    i += 1

X = np.array(X)
Y = np.array(Y)

print(X.shape)
print(Y.shape)
```

```
(10000, 96, 96, 3)
(10000,)
```

In [13]:
```
i = 77
print(Y[i])
plt.imshow(X[i])
```

```
0
```

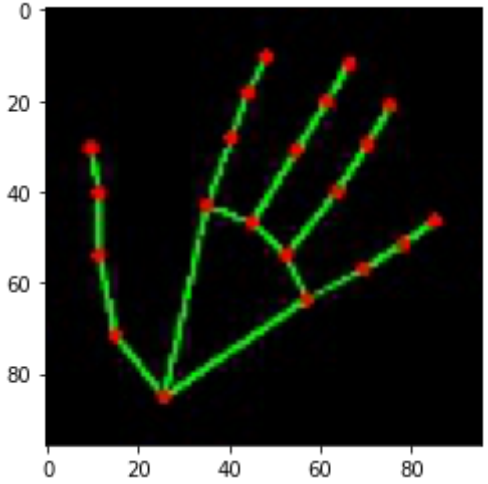
Out[13]: <matplotlib.image.AxesImage at 0x1942bf41388>



In [14]:
```
i = 1999
print(Y[i])
plt.imshow(X[i])
```

```
0
```

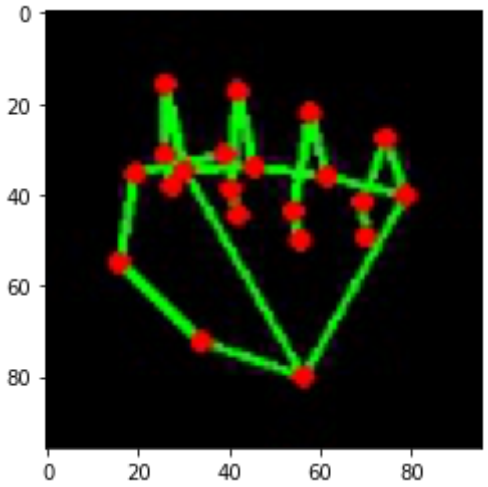
Out[14]: <matplotlib.image.AxesImage at 0x1942bedeac8>



In [15]:
```
i = 2700
print(Y[i])
plt.imshow(X[i])
```

```
1
```

Out[15]:
```
<matplotlib.image.AxesImage at 0x1943c742508>
```



In [16]:
```
X = X / 255
```

In [17]:
```
X[56]
```

Out[17]:
```
array([[[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],

       ...,

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],
```

```
[[0., 0., 0.],
 [0., 0., 0.],
 [0., 0., 0.],
 ...,
 [0., 0., 0.],
 [0., 0., 0.],
 [0., 0., 0.]]])
```

In [18]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 20, shuffle = True)
```

In [19]:

```
cnn = models.Sequential([
    #conv layer 01
    layers.Conv2D(input_shape = X.shape[1:], #layer 01
                  filters = 32,
                  kernel_size = (3,3),
                  strides = (1, 1),
                  padding = 'same',
                  activation = 'relu'),

    #maxpool layer 01
    layers.MaxPooling2D(pool_size = (2, 2), #layer 02
                        strides = (2,2),
                        padding = 'same'),

    #dropout layer 01
    layers.Dropout(0.2), #layer 04

    #conv layer 02
    layers.Conv2D(input_shape = X.shape[1:], #layer 05
                  filters = 64,
                  kernel_size = (3,3),
                  strides = (1, 1),
                  padding = 'same',
                  activation = 'relu'),

    #maxpool layer 02
    layers.MaxPooling2D(pool_size = (2, 2), #layer 06
                        strides = (2,2),
                        padding = 'same'),

    #dropout layer 02
    layers.Dropout(0.2), #layer 07

    #input layer
    layers.Flatten(), #layer 08

    #hidden layer 01
    layers.Dense(units = 512, #layer 09
                 activation = 'relu'),

    #dropout layer 03
```

```
    layers.Dropout(0.2), #Layer 10

    #output layer
    layers.Dense(units = len(input_types), #layer 11
                 activation = 'softmax')
])
```

In [20]:
```
cnn.compile(optimizer = 'adam',
            loss = 'sparse_categorical_crossentropy',
            metrics = ['accuracy'])
```

In [21]:
```
start_time = time.time()
cnn.fit(X_train, Y_train,epochs = 10)
end_time = time.time()
print("total time in seconds", (end_time - start_time))
```

```
Train on 8000 samples
Epoch 1/10
8000/8000 [==============================] - 18s 2ms/sample - loss: 0.1218 - accuracy: 0.9621
Epoch 2/10
8000/8000 [==============================] - 5s 612us/sample - loss: 0.0138 - accuracy: 0.9952
Epoch 3/10
8000/8000 [==============================] - 4s 562us/sample - loss: 0.0061 - accuracy: 0.9975
Epoch 4/10
8000/8000 [==============================] - 4s 516us/sample - loss: 0.0063 - accuracy: 0.9984
Epoch 5/10
8000/8000 [==============================] - 4s 514us/sample - loss: 0.0082 - accuracy: 0.9974
Epoch 6/10
8000/8000 [==============================] - 4s 516us/sample - loss: 0.0012 - accuracy: 0.9996: 0s - loss: 9
Epoch 7/10
8000/8000 [==============================] - 4s 514us/sample - loss: 0.0012 - accuracy: 0.9998
Epoch 8/10
8000/8000 [==============================] - 4s 511us/sample - loss: 0.0059 - accuracy: 0.9977
Epoch 9/10
8000/8000 [==============================] - 4s 510us/sample - loss: 0.0049 - accuracy: 0.9983
Epoch 10/10
8000/8000 [==============================] - 4s 507us/sample - loss: 4.8045e-04 - accuracy: 1.0000
total time in seconds 55.98964047431946
```

In [22]:
```
cnn.evaluate(X_test, Y_test)
```

```
2000/2000 [==============================] - 1s 347us/sample - loss: 0.0203 - accuracy: 0.9950
```

Out[22]:
```
[0.02028128568033543, 0.995]
```

In [23]:
```
Y_pred = cnn.predict(X_test)
Y_pred_classes = [np.argmax(e) for e in Y_pred]

print("Classification Report: \n", classification_report(Y_test, Y_pred_classes))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       399
           1       1.00      1.00      1.00       408
           2       0.99      1.00      0.99       373
           3       1.00      0.99      0.99       411
           4       1.00      0.99      0.99       409

    accuracy                           0.99      2000
   macro avg       0.99      1.00      0.99      2000
weighted avg       1.00      0.99      0.99      2000
```

In [24]:
```
cnn.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 96, 96, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 48, 48, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 48, 48, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 48, 48, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 24, 24, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 36864)             0
_________________________________________________________________
dense (Dense)                (None, 512)               18874880
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 2565
=================================================================
Total params: 18,896,837
Trainable params: 18,896,837
Non-trainable params: 0
_________________________________________________________________
```

In [25]:
```
cnn.get_weights()
```

Out[25]:
```
[array([[[[-4.28520143e-02,  4.36444543e-02,  1.59608930e-01,
           1.85733419e-02, -1.76627964e-01, -5.17254472e-02,
           1.54973298e-01,  1.52638137e-01, -5.96843921e-02,
           9.16987583e-02, -7.56276622e-02,  1.28879249e-01,
          -1.04584612e-01,  1.06672294e-01, -3.59708071e-03,
          -2.83434894e-02, -5.81433587e-02, -1.46537960e-01,
          -1.40992641e-01,  7.00010434e-02, -1.15938611e-01,
           1.51752919e-01,  4.53807190e-02, -3.70776281e-02,
           2.67298166e-02,  1.20139748e-01, -2.12287139e-02,
           5.76514937e-03,  3.40608172e-02,  2.66641732e-02,
           9.95498225e-02,  3.47049385e-02],
         [ 1.24757126e-01,  2.18364112e-02, -1.07748955e-01,
          -1.67609006e-01,  6.19483255e-02,  1.07955396e-01,
           1.45916596e-01,  8.77300203e-02, -4.40291055e-02,
          -3.93620431e-02,  8.53522345e-02, -7.14429617e-02,
          -4.09310609e-02,  1.11101434e-01, -6.19888566e-02,
           1.34559140e-01,  1.45799443e-01, -9.37778577e-02,
           4.06562909e-02,  4.98728603e-02,  1.39217839e-01,
           1.69269502e-01,  1.08848564e-01, -8.74814466e-02,
          -8.37267097e-03, -9.63342637e-02,  2.84434427e-02,
           1.16600014e-01, -1.46336481e-01,  2.22849566e-02,
          -5.36732487e-02,  1.41608968e-01],
         [-4.85812873e-03,  5.51559143e-02, -4.77876514e-02,
           6.68440089e-02, -1.92726299e-01,  1.39218450e-01,
           2.82772500e-02,  1.13179721e-01,  1.89193320e-02,
           1.27390832e-01, -5.07281013e-02,  2.93888021e-02,
           6.15407787e-02,  1.15959113e-02, -1.31116033e-01,
           1.02262676e-01, -5.39021902e-02, -4.15763482e-02,
           1.48542196e-01, -6.26639128e-02, -9.96292308e-02,
           1.26438931e-01,  5.79981506e-02,  7.03374892e-02,
           1.54981017e-01, -5.81746027e-02,  1.21636979e-01,
           4.84964736e-02,  1.07926615e-02,  1.03315562e-01,
          -5.19787893e-02, -3.22834635e-03]],

        [[-1.57409176e-01, -1.10887028e-01,  1.19368821e-01,
           4.24062982e-02, -1.54515788e-01, -7.97641948e-02,
          -7.76617825e-02, -1.35561332e-01,  6.85818344e-02,
           3.29237059e-02, -3.63977291e-02,  1.15845680e-01,
           8.99454951e-02,  1.59379140e-01, -7.30772838e-02,
           6.14315160e-02,  1.69435889e-02, -1.10391386e-01,
          -6.85885549e-02,  1.84984967e-01, -1.23387843e-01,
```

```
   3.92439254e-02,  2.95328014e-02,  5.05197560e-03,
  -6.68328851e-02, -3.71791283e-03,  1.50715441e-01,
   8.70548040e-02, -1.29327103e-01, -4.76575755e-02,
   1.53815359e-01, -1.18578836e-01],
 [ 3.95396724e-03,  5.49856797e-02, -3.19713764e-02,
  -1.89749181e-01,  9.24612135e-02,  7.00220317e-02,
  -5.25212027e-02,  2.02823170e-02, -6.83283284e-02,
  -1.01530634e-01, -7.31651932e-02, -1.83818936e-01,
   1.16775535e-01,  1.75797287e-03, -1.22805648e-01,
   1.22662149e-01, -8.30495283e-02,  3.93665098e-02,
   4.27106731e-02,  1.03711352e-01,  1.20188475e-01,
  -1.55766541e-02, -1.03493139e-01, -7.54922107e-02,
   3.37243229e-02, -5.19191884e-02, -5.54912984e-02,
  -2.04783194e-02, -1.65224344e-01, -4.70946431e-02,
   1.97476638e-03, -3.46633159e-02],
 [-1.24302894e-01, -1.82111248e-01,  5.62388822e-02,
  -1.25474408e-01, -2.56664045e-02, -3.60028110e-02,
  -5.74664772e-02,  1.01127021e-01, -4.74841893e-02,
   3.94921489e-02, -6.02993332e-02, -7.03132451e-02,
   1.41236484e-01,  9.56109166e-02, -9.35035050e-02,
   9.01390463e-02,  3.66556644e-03, -9.60493162e-02,
   1.17124371e-01, -2.85387579e-02, -3.10916454e-02,
   1.44075528e-01,  8.30006152e-02,  1.10915698e-01,
   1.67114120e-02, -4.11828384e-02,  1.38709366e-01,
   7.67610222e-02, -8.82508755e-02, -3.83544108e-03,
   3.02454410e-03,  8.96851812e-03]],

[[-1.22680843e-01,  9.29156691e-02,  1.24032125e-01,
  -1.56349540e-01, -5.80371395e-02,  2.47790609e-02,
   1.28605917e-01, -1.31432980e-01,  1.17688417e-01,
   3.54333185e-02, -7.29179084e-02, -1.07030243e-01,
   3.67022939e-02,  1.38840392e-01,  6.88146949e-02,
  -1.27811000e-01, -1.14382610e-01, -4.95465659e-03,
   7.00943321e-02,  1.52391613e-01, -8.78099278e-02,
  -1.55963928e-01,  5.30446768e-02, -2.06147525e-02,
  -1.37059465e-02,  1.31358564e-01,  8.29909071e-02,
   6.24938160e-02,  2.52410211e-02, -7.90996626e-02,
   9.80370492e-02,  1.44432366e-01],
 [-9.93269235e-02, -1.64214462e-01,  4.59141620e-02,
  -9.48083848e-02,  1.26117868e-02,  7.32767805e-02,
   8.88923332e-02, -1.50267988e-01, -3.89665328e-02,
   1.30762413e-01,  1.05070196e-01,  1.45238355e-01,
```

```
   -8.05041790e-02, -1.28127173e-01, -7.56547078e-02,
    8.64235610e-02, -4.24838737e-02, -2.99068488e-04,
    1.06451415e-01, -3.68188657e-02, -7.96692595e-02,
   -1.48392111e-01, -1.28999978e-01,  9.67338681e-02,
    3.46082039e-02, -5.55897467e-02,  8.63774344e-02,
   -1.42554089e-01, -1.25038788e-01,  2.72523500e-02,
   -1.62237376e-01,  5.98995201e-02],
  [ 1.17753766e-01,  7.60909021e-02,  1.31999582e-01,
   -9.08713974e-03, -6.18140586e-02,  8.04050788e-02,
   -1.29936393e-02, -3.12164351e-02, -9.68077108e-02,
   -3.54661280e-03, -6.46771863e-02,  8.92770812e-02,
    5.36139943e-02, -1.71392322e-01, -3.84497792e-02,
   -1.08272992e-02,  7.81316832e-02, -1.25815630e-01,
    2.20573856e-03, -7.72438198e-02, -5.78110442e-02,
   -4.99096997e-02, -3.65810394e-02, -4.70468439e-02,
   -7.24415928e-02,  3.82824279e-02,  1.85186490e-01,
   -9.63366479e-02, -1.01434812e-01,  8.67642462e-03,
    2.00230274e-02,  6.49010064e-03]]],


[[[ 1.56864598e-01,  1.51893914e-01, -1.44517019e-01,
    1.12386309e-01,  8.14731419e-02,  2.02344973e-02,
    8.27356279e-02,  1.22515790e-01,  8.90079420e-03,
   -1.76305324e-02,  7.09909648e-02, -4.76421043e-02,
    8.54269564e-02,  1.46211639e-01, -9.48653147e-02,
    4.02112678e-02,  8.39474201e-02, -6.58185780e-02,
   -6.06778217e-03, -1.22044213e-01, -6.08186796e-02,
    3.68744470e-02, -1.27556667e-01,  1.33689255e-01,
    5.03318012e-02, -1.04411200e-01, -1.97228212e-02,
   -8.67015719e-02,  1.79675251e-01,  4.67115790e-02,
    1.29238605e-01, -1.08463846e-01],
  [ 2.11005379e-03,  7.79224783e-02, -3.61965187e-02,
   -4.63297553e-02,  1.83684919e-02,  1.08603410e-01,
    2.00597197e-02, -1.04563661e-01, -1.84665609e-03,
   -1.06990024e-01,  1.73229296e-02,  6.73637390e-02,
   -1.26746356e-01, -5.83564378e-02, -4.35363241e-02,
   -6.99516386e-02, -8.73355381e-03,  6.99344277e-02,
    1.01653017e-01,  1.08869128e-01, -1.02209657e-01,
   -1.03588082e-01,  1.31505951e-01,  9.20896903e-02,
    1.76718619e-04, -1.07962877e-01,  9.03707519e-02,
    4.84215468e-02, -9.59647000e-02,  6.67123497e-02,
   -4.48843613e-02, -6.96909502e-02],
```

```
 [ 1.00330308e-01,  1.11114666e-01, -1.45347640e-02,
   9.46790352e-02, -8.83034393e-02,  4.69644219e-02,
   1.02333426e-01, -2.77465712e-02,  7.84563739e-03,
   7.02427700e-02, -3.97369787e-02,  1.30657583e-01,
   3.50758084e-04, -3.72191742e-02,  1.20221116e-01,
   1.20657898e-01,  7.17914104e-02, -1.13882363e-01,
   9.44648683e-03,  9.45191085e-02, -2.14922559e-02,
   1.69780699e-03,  1.51151508e-01,  1.13324456e-01,
  -1.02751236e-03,  5.41279744e-03,  3.28579023e-02,
  -3.81403379e-02, -1.26237243e-01,  4.45276350e-02,
   7.60251889e-03, -7.54458457e-02]],

[[ 4.46881205e-02,  3.45630050e-02, -2.54760273e-02,
  -1.00111894e-01,  2.05307221e-03, -9.09524113e-02,
   1.49255216e-01,  4.41162698e-02, -4.28332202e-02,
   5.50987199e-03, -1.02641344e-01, -8.47098306e-02,
   4.22903337e-02, -3.22911236e-03,  1.06394954e-01,
  -1.75006330e-01, -1.45461308e-02, -1.27740800e-01,
  -1.49168074e-01,  6.95187375e-02, -1.21643424e-01,
   1.19282342e-01,  6.48809522e-02, -2.87600458e-02,
  -2.68885940e-02,  4.71640229e-02, -8.43802094e-02,
   2.60377210e-02, -1.07900023e-01,  8.37432519e-02,
   1.41519353e-01, -4.88128290e-02],
 [ 1.58930924e-02, -1.34825051e-01, -2.27533542e-02,
  -1.65982783e-01, -7.52273574e-02,  6.86361194e-02,
  -5.23933992e-02,  1.23333901e-01, -7.50305876e-02,
  -1.28882200e-01,  9.26023424e-02, -1.40563399e-01,
   3.34331095e-02,  1.24304565e-02,  5.42983972e-02,
   1.32043973e-01, -1.11680150e-01,  1.18369676e-01,
   9.74285379e-02, -4.43356559e-02, -3.94948246e-03,
  -5.09889983e-02, -1.11098625e-01,  6.13604411e-02,
  -1.68498993e-01,  2.35631615e-02,  3.03343479e-02,
   3.19847129e-02, -4.49926443e-02,  1.50461733e-01,
  -5.71797527e-02,  1.03000656e-01],
 [ 8.40079132e-03,  1.98631436e-02, -4.52392399e-02,
   6.83390945e-02,  8.78653452e-02,  1.04821406e-01,
  -3.33126634e-02,  4.19680662e-02,  8.50611329e-02,
  -3.75258066e-02,  7.41654038e-02, -5.76567315e-02,
   1.15431780e-02, -3.92206609e-02,  1.00367002e-01,
   5.89563362e-02,  7.44001046e-02, -2.16415580e-02,
  -1.04689151e-01,  1.00606307e-01, -1.23090193e-01,
   9.16149542e-02,  7.65650300e-03,  9.41911414e-02,
```

```
   -1.66297302e-01, -9.11236927e-02,  2.63381451e-02,
    3.43099311e-02, -5.65537252e-02,  9.35591310e-02,
   -7.09670112e-02,  3.27264480e-02]],

 [[ 9.61344615e-02,  2.60419082e-02,  7.70430788e-02,
    2.55581327e-02, -8.81875604e-02, -1.50934801e-01,
    1.23130336e-01, -4.26980332e-02,  1.86082587e-01,
   -8.74019191e-02,  3.17253582e-02, -8.74888524e-02,
    5.58330566e-02,  4.42438200e-03, -1.01260990e-01,
    3.43621150e-02, -8.02746043e-02,  1.00454204e-01,
   -1.61379039e-01, -1.96860097e-02, -1.75340474e-01,
   -1.23112962e-01, -6.04524985e-02,  8.47493336e-02,
   -3.16696987e-02, -1.47429658e-02, -1.31647423e-01,
    8.10297877e-02,  5.07490002e-02, -2.64045559e-02,
   -7.58418366e-02,  1.02166437e-01],
  [ 1.17264383e-01, -7.32460842e-02,  3.86406295e-02,
    1.95346139e-02, -9.85118188e-03, -1.06822550e-01,
   -9.62382331e-02, -1.21342383e-01, -1.21950187e-01,
   -7.78447315e-02,  1.29141837e-01,  6.38256669e-02,
   -1.17137879e-01,  1.69113744e-03,  9.13009271e-02,
    1.37447894e-01,  1.00738585e-01, -7.83271119e-02,
   -1.07124625e-02,  9.49226543e-02,  1.17423639e-01,
    1.14311822e-01, -6.56054989e-02,  1.02939382e-01,
   -1.36812210e-01,  4.40323614e-02, -7.24631399e-02,
   -1.49592862e-01, -1.77853942e-01,  1.01824813e-01,
   -9.46110208e-03,  5.71540408e-02],
  [ 1.22882918e-01, -1.35526463e-01,  1.88621972e-02,
   -4.69882488e-02,  1.36000127e-01,  4.56308834e-02,
   -7.68457651e-02, -4.15249132e-02,  1.04640000e-01,
    3.16003673e-02, -4.91624027e-02,  1.11729734e-01,
   -1.07953407e-01, -1.13240696e-01,  2.03532889e-01,
    2.55311280e-02,  8.24333206e-02, -1.31063506e-01,
    8.21234565e-03, -1.08545266e-01, -8.69535729e-02,
   -1.01924539e-02, -4.63425741e-02,  3.36958538e-03,
   -1.32384181e-01,  1.52742773e-01, -5.66399544e-02,
    6.07990660e-02,  1.36402756e-01,  1.31102905e-01,
   -1.03456490e-01, -9.57960077e-03]]],


[[[ 1.56354249e-01,  4.08216566e-02,  5.11618890e-02,
    1.28537342e-01,  1.46580905e-01,  1.40429944e-01,
    4.46219817e-02, -9.39729363e-02, -1.11674055e-01,
```

```
  -1.11851692e-01,  4.58808616e-02, -8.61246884e-02,
   6.42067790e-02, -1.72654428e-02, -8.02289844e-02,
  -1.00180343e-01,  1.14738248e-01,  1.66967154e-01,
  -8.21605921e-02,  1.10419272e-02, -9.33586732e-02,
   3.18632945e-02, -1.37285233e-01,  5.79681955e-02,
  -5.00628687e-02, -1.20908424e-01, -2.63227280e-02,
   1.86727159e-02,  1.53086960e-01, -6.45490661e-02,
   1.56057775e-01, -1.48420915e-01],
 [-1.21922083e-01, -3.02628577e-02,  6.58548549e-02,
  -3.03998422e-02,  3.67683582e-02,  1.16760701e-01,
   1.03525184e-01,  2.66533140e-02,  4.13296968e-02,
  -1.87324509e-01,  2.16424130e-02,  2.40287520e-02,
  -1.01134643e-01,  1.00959733e-01,  9.37981829e-02,
  -1.48516586e-02, -6.96254820e-02,  8.34217966e-02,
   1.19632026e-02, -1.55773252e-01,  1.43380016e-01,
   8.02232772e-02,  4.36313599e-02,  9.77273798e-04,
   5.07352240e-02, -1.79292232e-01,  6.38395548e-02,
   1.44502014e-01, -5.52834943e-02,  8.80018026e-02,
  -6.94827437e-02, -1.34461612e-01],
 [ 5.97560965e-02,  1.30014047e-02, -3.96946371e-02,
   7.69917145e-02,  4.31287065e-02,  4.95051965e-02,
   2.00897194e-02, -8.14439431e-02,  6.76162615e-02,
  -1.10980667e-01, -1.09577723e-01,  6.38301810e-03,
  -4.26838882e-02, -1.69892819e-03, -1.01334825e-01,
   8.33152980e-02, -5.95281869e-02, -5.54894283e-02,
   1.00199319e-01, -8.00248906e-02, -3.84541228e-02,
  -4.20915410e-02, -3.43922600e-02, -6.11651503e-02,
  -7.31993467e-02, -4.10636961e-02,  7.00724721e-02,
  -7.55806938e-02, -1.04597665e-01,  3.36056463e-02,
   6.81572035e-02, -5.29801697e-02]],

[[-1.32437259e-01,  1.08375147e-01, -1.75532922e-02,
   9.63248834e-02,  5.26133180e-02,  8.66120160e-02,
   5.80953918e-02, -1.33253992e-01,  1.42692864e-01,
   3.73363830e-02,  9.66844410e-02, -1.52375586e-02,
   1.32831559e-01,  1.01945922e-01, -1.51124611e-01,
  -1.47702351e-01,  1.63243040e-01,  1.34520277e-01,
   8.61110538e-02, -6.53339028e-02,  4.50073071e-02,
  -6.90295734e-03, -2.68882867e-02,  1.13443226e-01,
  -1.61924675e-01, -7.26378784e-02,  2.13642381e-02,
   2.92498711e-02,  9.73007306e-02, -1.26667425e-01,
   1.39086306e-01, -2.61375792e-02],
```

```
 [-2.20210105e-02, -7.32498094e-02,  6.19876720e-02,
  -6.33684099e-02, -4.86947969e-02, -1.53756037e-01,
   6.78447410e-02, -8.28900859e-02,  1.03702977e-01,
  -4.13871780e-02, -1.79075941e-01,  2.67204028e-02,
   5.36019076e-03,  8.82540345e-02,  5.24277054e-02,
   9.12201479e-02, -1.10492520e-01, -7.63486102e-02,
   1.59527883e-02,  9.92905535e-03,  1.92591846e-01,
  -1.73449695e-01, -1.06260134e-02,  1.08209454e-01,
   6.97546378e-02, -1.85547486e-01, -9.33744982e-02,
  -9.60157439e-02,  2.82840095e-02,  2.25283001e-02,
   1.63352047e-03, -8.17042813e-02],
 [-5.11195734e-02, -4.40096073e-02, -3.07718068e-02,
  -8.07226971e-02,  9.52127948e-02, -5.32693043e-02,
   2.61848643e-02, -1.17748775e-01,  3.14401388e-02,
  -3.59238982e-02, -1.18763842e-01, -1.34754539e-01,
  -1.21216208e-01,  7.56445294e-03, -9.03730560e-03,
  -2.98859496e-02, -1.45079032e-01, -1.17432155e-01,
  -5.98382391e-03, -1.71585694e-01, -4.17567007e-02,
  -6.66006431e-02, -5.45995459e-02,  1.99080668e-02,
  -1.55349106e-01,  3.97651866e-02, -1.37927204e-01,
   2.66157510e-03, -1.26399279e-01, -1.34121507e-01,
  -3.94530743e-02, -3.05562746e-02]],

[[-5.71134463e-02,  1.00360781e-01, -1.83406055e-01,
   3.03677246e-02,  7.32002556e-02,  1.11305363e-01,
   1.88194420e-02,  2.04492211e-02,  1.72158688e-01,
  -1.45716578e-01, -9.62185636e-02,  8.98498893e-02,
   1.39427543e-01, -1.12953968e-01, -1.43095359e-01,
  -1.04722932e-01, -1.01959378e-01,  2.79424414e-02,
   1.81722622e-02,  3.56778912e-02, -9.50722545e-02,
  -2.26660911e-02,  2.47096512e-02,  1.72102377e-02,
  -8.80867708e-03,  6.51854053e-02, -1.00731485e-01,
  -2.47257594e-02,  4.39742431e-02, -8.32502171e-02,
   4.82753739e-02, -1.23993546e-01],
 [-3.87790017e-02,  7.20745921e-02,  1.07049346e-01,
   1.23272270e-01, -1.21151298e-01, -1.28531069e-01,
  -2.83657219e-02,  1.25277176e-01, -4.96240556e-02,
  -1.72417641e-01,  1.26892403e-01,  1.08451052e-02,
   1.02589056e-01, -1.40129745e-01,  1.30274296e-01,
  -7.86730368e-03, -2.65162047e-02,  3.60450372e-02,
  -1.19161427e-01, -9.49309617e-02,  4.75128256e-02,
  -4.23426107e-02,  1.16918728e-01,  9.80549771e-03,
```

```
          -1.57681853e-01,  4.49857712e-02, -2.88898647e-02,
          -7.61553869e-02,  3.47784385e-02,  1.17743965e-02,
          -1.40468821e-01, -1.91960149e-02],
         [-7.00177401e-02, -3.97073850e-02,  8.89279619e-02,
          -1.07368417e-01, -1.10147111e-01,  8.37447569e-02,
          -1.37647152e-01,  7.46508166e-02, -1.70943048e-02,
           9.81924385e-02, -1.75582558e-01, -1.24945931e-01,
           2.78596040e-02,  4.83250655e-02, -1.67997181e-02,
          -1.26743868e-01, -3.27588506e-02,  6.62139431e-02,
          -1.02039380e-02, -7.01699127e-03, -2.96668243e-02,
           9.82268974e-02,  4.83106486e-02,  7.01504424e-02,
           8.16611797e-02, -2.20034160e-02, -3.02171148e-02,
           9.75378789e-03,  6.32397458e-02,  8.47995132e-02,
           9.19961482e-02,  4.91685644e-02]]]], dtype=float32),
array([-0.01444867, -0.02897281, -0.04490011, -0.03394422, -0.01035008,
       -0.02165564, -0.05194209, -0.01358057, -0.03425884, -0.01882469,
       -0.05161554, -0.0310229 , -0.01701346, -0.05634074, -0.02388675,
       -0.03986982,  0.00369636, -0.0303822 , -0.04299281, -0.03385617,
        0.00622817, -0.02944829, -0.02333611, -0.0457902 , -0.03227441,
       -0.01522297, -0.03319498, -0.03275938, -0.01180827, -0.04529081,
       -0.0319908 , -0.01652381], dtype=float32),
array([[[[-0.01963853, -0.06624183,  0.07816555, ...,  0.02276716,
           0.03688964,  0.06632795],
         [ 0.04389834, -0.03747228, -0.05481068, ..., -0.04420754,
           0.06494352,  0.12065294],
         [-0.04330111, -0.11930034,  0.04000943, ...,  0.00066422,
          -0.06434234,  0.07471962],
         ...,
         [-0.03306649,  0.03198174,  0.00808297, ...,  0.00082465,
           0.07110166, -0.05346214],
         [-0.08700914, -0.02589184, -0.06131515, ..., -0.04760548,
           0.01707066,  0.13059554],
         [-0.04892919,  0.00551735,  0.06560505, ...,  0.01473016,
          -0.03778651, -0.02850114]],

        [[ 0.03925117, -0.03287424, -0.02703222, ..., -0.09549502,
           0.06891216,  0.00141086],
         [-0.02586645, -0.09964671, -0.02745368, ..., -0.03146445,
          -0.0530087 ,  0.0561355 ],
         [-0.09573913,  0.00032605,  0.0093409 , ...,  0.01485541,
           0.05412714,  0.03250152],
         ...,
```

```
  [-0.01797   ,  0.00676979,  0.09902436, ..., -0.08120924,
   -0.00104912,  0.05125952],
  [ 0.03233477, -0.04871088, -0.04726516, ..., -0.05900403,
   -0.02335955,  0.01783145],
  [-0.09767219,  0.03381046,  0.01689845, ...,  0.05910322,
    0.02592258, -0.00749721]],

 [[-0.03669447, -0.04489919, -0.01027058, ..., -0.1112963 ,
    0.10468107, -0.0915991 ],
  [-0.04129813, -0.10182003,  0.01531142, ..., -0.09693678,
   -0.05743657,  0.05951152],
  [ 0.00032306, -0.0553063 ,  0.04476694, ..., -0.10745019,
    0.00744299, -0.05842544],
  ...,
  [-0.03168156, -0.00231675,  0.02804218, ..., -0.0281569 ,
    0.04211229, -0.01132352],
  [-0.03630136, -0.08517024, -0.01043805, ..., -0.09077629,
   -0.09959631, -0.00446928],
  [-0.02501991,  0.00057854,  0.01456585, ..., -0.03024624,
    0.09144545,  0.01894899]]],


[[[-0.04438017,  0.04424624,  0.11423019, ..., -0.03784728,
   -0.05689745,  0.01617585],
  [ 0.05673828, -0.05066194,  0.00841722, ...,  0.02306619,
    0.07837442, -0.03543691],
  [ 0.05714456, -0.00598422, -0.01052115, ...,  0.01461733,
   -0.02784073,  0.08651435],
  ...,
  [ 0.03842506,  0.03585546,  0.03374446, ...,  0.0235522 ,
   -0.00451657, -0.09422679],
  [-0.09403548,  0.03194315,  0.00851833, ...,  0.0026237 ,
   -0.02532791,  0.01964636],
  [-0.07944588, -0.11965495,  0.01483067, ..., -0.06129063,
    0.02936094, -0.10776504]],

 [[ 0.09142108, -0.04499345,  0.08994363, ...,  0.04526019,
   -0.11516403, -0.00125523],
  [ 0.01215978, -0.08889228, -0.01211287, ...,  0.07245694,
   -0.02142784, -0.09483367],
  [-0.02010036,  0.05836384, -0.10839546, ..., -0.04536323,
   -0.02327878, -0.0029034 ],
```

```
  ...,
  [ 0.04506005, -0.06238261, -0.01281136, ...,  0.02680029,
   -0.06130828, -0.06586076],
  [ 0.06484816, -0.06264126, -0.01288765, ..., -0.06586061,
   -0.08287326, -0.02750882],
  [-0.11787965,  0.02109547,  0.00231362, ..., -0.06110713,
   -0.07186067,  0.00158714]],

 [[-0.00158415, -0.02182055, -0.07733681, ..., -0.07861228,
    0.08031048, -0.1294131 ],
  [-0.08289751,  0.06211606, -0.02456911, ..., -0.09752941,
   -0.06475592, -0.07896794],
  [-0.00462128, -0.07275952,  0.01943522, ...,  0.01536254,
   -0.03625599, -0.08483479],
  ...,
  [-0.08329564, -0.00823278, -0.05483642, ..., -0.01294453,
    0.05990432,  0.00502241],
  [-0.1157658 , -0.07090121,  0.05912757, ...,  0.00554302,
   -0.05947027, -0.05518537],
  [-0.02435072,  0.03736395, -0.05732189, ...,  0.00907481,
   -0.00036825, -0.09653358]]],


[[[ 0.01594588,  0.04692076, -0.01660899, ...,  0.09540597,
   -0.03827992, -0.08875856],
  [ 0.02880665,  0.02551271,  0.04056923, ...,  0.04053366,
   -0.03720813, -0.02695349],
  [ 0.02453976, -0.07363388, -0.02582048, ...,  0.07035876,
   -0.11513467, -0.03033928],
  ...,
  [-0.06439478,  0.01429694,  0.00662419, ..., -0.02704753,
   -0.15883012, -0.13930058],
  [-0.07795814, -0.04813858,  0.01732613, ...,  0.10150526,
    0.00367835,  0.09562478],
  [ 0.08880619, -0.08342678,  0.01159345, ..., -0.01169418,
   -0.12302651, -0.11479883]],

 [[ 0.00386888, -0.00155396, -0.0050973 , ...,  0.00750542,
    0.02596974, -0.00867824],
  [ 0.00668502, -0.10743056, -0.05827129, ...,  0.00449243,
   -0.11942431, -0.00723517],
  [ 0.06308983,  0.00132755, -0.00928753, ...,  0.02051499,
```

```
           0.0238309 ,  0.01231907],
         ...,
         [-0.03407032, -0.04600997, -0.04930986, ..., -0.04351307,
           0.00986414, -0.06928847],
         [ 0.04339812, -0.00956215, -0.04305512, ...,  0.09657822,
          -0.02245595,  0.0368919 ],
         [-0.02181758,  0.04877091,  0.01686   , ..., -0.10878512,
          -0.07552449, -0.04460378]],

        [[-0.09882949,  0.0222514 , -0.03584594, ...,  0.05010591,
           -0.1659322 ,  0.00606412],
         [ 0.04180552, -0.08253966, -0.05549619, ..., -0.04657116,
          -0.05640827,  0.10101113],
         [-0.11718663, -0.00864948, -0.02256193, ..., -0.12473156,
          -0.08445855, -0.01521849],
         ...,
         [-0.06581421, -0.11140288, -0.08820998, ..., -0.00527797,
           0.03273769, -0.11961377],
         [-0.07806088, -0.04405349, -0.08459641, ..., -0.0417449 ,
           0.03658207,  0.10988418],
         [-0.00560806, -0.08324367,  0.01082208, ..., -0.11311428,
          -0.01005983, -0.14685413]]]], dtype=float32),
 array([-0.01973484, -0.01530668, -0.02673298, -0.01592044, -0.02614121,
        -0.01570519, -0.01841594, -0.01927722, -0.01215652, -0.0134562 ,
        -0.02708036, -0.01868064, -0.02377446, -0.03296616, -0.01745386,
        -0.01565001, -0.02889147, -0.02890884, -0.04412843, -0.00756775,
        -0.01353511, -0.00448049, -0.0079375 , -0.03180924, -0.01727922,
        -0.01126407, -0.01366987, -0.02682471, -0.00740716, -0.00706726,
        -0.03564813, -0.02992731, -0.01527119, -0.01989275, -0.04571172,
        -0.01705603, -0.02420111, -0.03130183, -0.01787674, -0.0208976 ,
        -0.02143209, -0.00719035, -0.00443411, -0.01925581, -0.00713052,
        -0.03155063, -0.02757852, -0.01304793, -0.01212389, -0.03315372,
        -0.00675341, -0.00856029, -0.01038746, -0.01444477, -0.00884144,
        -0.00318988, -0.00850138, -0.02816829, -0.02509514, -0.03383898,
        -0.01871641, -0.02613041, -0.02383577, -0.00312279], dtype=float32),
 array([[-9.9900141e-03, -2.4115592e-03,  3.6659185e-03, ...,
          -4.2062718e-03,  8.4702419e-03, -7.2877575e-04],
        [-9.2192013e-03, -7.3209391e-03,  3.7364587e-03, ...,
          -8.2968809e-03, -1.0443784e-02,  1.9793529e-03],
        [-4.0204171e-04,  1.0537546e-02, -4.7742696e-03, ...,
           2.2787871e-03,  8.2946299e-03,  4.8363321e-03],
        ...,
```

```
        [-6.4190975e-03,  1.1791569e-02,  6.4278729e-03, ...,
          5.7635400e-03, -6.8204915e-03,  8.2735261e-03],
        [-1.0055332e-02, -3.1488338e-03,  5.4083960e-03, ...,
          1.5505487e-03, -1.3695949e-02,  1.2935912e-02],
        [ 2.8280951e-05,  8.2852300e-03,  1.3445616e-02, ...,
          8.8577624e-03,  4.6109874e-03, -1.1075396e-02]], dtype=float32),
array([-2.19771471e-02, -2.98628304e-03,  4.82374057e-03,  2.14951718e-03,
       -1.45814084e-02, -2.21666843e-02, -7.69605581e-03, -4.24563070e-04,
       -3.36677185e-03, -3.90766561e-03, -1.28643410e-02, -1.01687224e-03,
        8.58834572e-03,  7.50548905e-03,  1.32571021e-02,  1.26201585e-02,
       -1.37835247e-02, -6.01714302e-04, -2.87242862e-03, -3.05732456e-03,
        3.42693320e-03, -1.47321895e-02,  1.49313482e-02, -4.45319107e-03,
       -6.00501196e-03, -7.00138556e-03,  1.65054910e-02, -5.32461936e-03,
       -4.57534520e-03, -2.67642172e-04, -6.00426644e-03, -6.00452861e-03,
        9.51481925e-05, -5.37482137e-03, -2.89325463e-03, -6.00438425e-03,
       -5.01637161e-03, -3.05504277e-02,  6.04617177e-03, -1.12426551e-02,
       -6.61375327e-03, -6.41799252e-03, -6.36679959e-03, -9.70001239e-03,
       -6.08715881e-03, -2.73137446e-03, -6.00263802e-03, -9.57670715e-03,
        1.08657861e-02, -6.78049773e-03,  1.98754724e-02, -3.88146238e-03,
       -4.59731976e-03,  5.07779745e-03, -6.00421522e-03, -1.49858661e-03,
       -5.99990739e-03,  5.03354659e-03, -1.21418796e-02,  6.24262309e-03,
       -2.01757941e-02,  2.20351643e-03, -4.94568725e-04, -8.58450960e-03,
       -6.14087190e-03, -2.85741910e-02, -6.00145571e-03, -5.77422557e-03,
        4.82713059e-03,  1.30540067e-02, -7.07889674e-03, -3.45528126e-03,
        1.01868454e-02, -1.06234904e-02, -7.62192532e-03, -2.45134137e-03,
       -3.25194001e-03, -5.59036527e-03,  1.02721509e-02, -4.25552297e-03,
       -1.10975457e-02, -4.01557004e-03,  4.82645957e-03, -3.31792119e-03,
       -7.16382731e-03, -4.38707508e-03,  3.20946286e-03,  7.50046875e-03,
       -6.00083172e-03, -5.00462484e-03, -3.69552686e-03, -9.86704137e-03,
       -2.88403649e-02, -1.25794420e-02, -2.17061886e-03, -7.21456110e-03,
       -9.28298663e-03, -6.54980913e-03,  1.01177292e-02,  9.47497785e-03,
       -4.44248458e-03,  4.02076449e-03, -1.06326370e-02, -6.00391906e-03,
        5.94671723e-03, -4.77222307e-03, -6.17146399e-03,  2.03409027e-02,
       -1.23699233e-02, -8.67931079e-03, -4.40996839e-03, -7.12397508e-03,
       -1.15637453e-02, -3.72816168e-04, -3.21114040e-03, -4.90393862e-03,
       -4.73849196e-03, -1.29645057e-02, -1.90549362e-02, -6.00273069e-03,
        1.11949667e-02, -6.00129087e-03, -1.79670528e-02, -3.61489225e-03,
       -2.61768140e-03, -6.00376679e-03, -2.19061114e-02, -4.98636719e-03,
       -6.00322429e-03, -1.09725865e-03, -1.62695181e-02, -6.06374815e-03,
       -6.00199262e-03, -7.43175857e-03,  6.95080450e-03, -6.00457238e-03,
       -6.00430416e-03, -4.21322742e-03,  8.43355153e-03, -4.11126502e-02,
       -2.28148163e-03, -2.53218878e-02,  1.43373981e-02,  1.01334527e-02,
```

```
 2.43720436e-03, -3.84591683e-03, -6.00446528e-03, -1.04980292e-02,
-5.55506162e-03,  4.30358201e-03,  1.45403640e-02,  3.55922221e-03,
-2.98163481e-02,  1.60100730e-03, -8.66092741e-03, -2.52882708e-02,
-9.81309265e-03, -1.71480104e-02, -6.00311579e-03, -1.37755582e-02,
-4.57183970e-03,  7.39659276e-03,  1.35252206e-02, -6.84603490e-03,
-1.24814687e-02, -6.00396376e-03, -1.66728022e-03, -6.00364665e-03,
 3.93993780e-03, -1.72974877e-02, -1.11181494e-02, -2.15390529e-02,
-1.12146670e-02, -7.36748101e-03, -1.10178767e-02,  3.80296074e-03,
-4.94363485e-03, -5.98829193e-03,  1.01786964e-02, -9.14536603e-03,
 4.87482408e-03,  2.44819280e-03,  4.25296044e-03,  1.14966556e-02,
 6.38313591e-03, -4.06416319e-03, -1.59612428e-02,  1.14766536e-02,
-3.79890134e-03,  3.75549169e-03,  1.98880117e-02, -2.25125458e-02,
-4.45554219e-02,  5.29494276e-03, -9.41743236e-03, -2.32668524e-03,
-6.00249553e-03,  2.66495440e-03, -6.00304082e-03, -9.34200641e-03,
 7.99611863e-03, -6.00331742e-03,  2.11192775e-04,  9.31882206e-03,
-4.02567210e-03,  2.41174735e-02, -4.87691537e-03,  1.17221987e-02,
 6.12396654e-03, -6.00163871e-03, -5.99947944e-03, -1.38028767e-02,
-7.79359182e-03, -4.14745323e-03, -1.59764942e-02,  9.86460410e-03,
-1.30829355e-02, -7.00038532e-03, -1.29517326e-02, -1.13453362e-02,
 1.16147986e-03, -5.99975139e-03, -2.30758209e-02,  1.10825179e-02,
-4.09746636e-03, -8.51876568e-03, -6.00193115e-03, -3.99336359e-03,
-9.78575274e-03, -6.00438099e-03,  2.54709125e-02, -2.65014777e-03,
-1.23502742e-02, -6.00225246e-03,  1.18637364e-02, -6.00444665e-03,
-6.49823667e-03, -3.57469474e-03, -3.77099239e-03, -6.75471791e-04,
-6.00405922e-03, -5.99638326e-03,  4.79686493e-03, -1.55934682e-02,
-1.34249148e-03, -6.00396935e-03, -5.50504122e-03, -2.65419651e-02,
 6.38829963e-03, -4.43961145e-03, -5.95776364e-03, -2.13987427e-03,
-8.33768025e-03, -1.90066248e-02, -7.86664616e-03, -6.00431534e-03,
-6.00416306e-03,  1.53589097e-03,  7.51623744e-03, -4.17802762e-03,
-3.51987453e-03, -7.56001705e-03, -3.01291817e-03, -5.32830041e-03,
-1.70402683e-03,  2.09116228e-02,  1.98598672e-02, -6.00403361e-03,
 2.40555406e-02, -7.51848420e-05, -6.00375887e-03, -1.12220682e-02,
-6.00350089e-03,  1.88643914e-02, -6.00218726e-03, -3.77116492e-03,
 8.66912480e-04, -1.97010092e-03, -6.00324851e-03, -1.25305662e-02,
 8.67653638e-03,  7.16986181e-03,  5.26185753e-03, -5.98573592e-03,
-8.86865798e-03, -6.00384269e-03, -1.11608468e-02,  3.80944740e-03,
-8.44237767e-03, -3.13059473e-03,  9.87399276e-03, -1.22110471e-02,
-6.00132300e-03,  7.98475463e-03, -6.76790019e-03, -1.81463966e-03,
-3.66186211e-03, -7.57851172e-04, -6.00465247e-03, -7.53932446e-03,
-5.98415453e-03, -1.41398683e-02, -6.00900687e-03, -6.00343896e-03,
 7.40362145e-03, -6.06807088e-03,  1.36094214e-02, -1.19422842e-02,
-7.21927965e-03, -5.01072593e-03,  1.25618400e-02,  3.15910927e-03,
```

```
-7.43998494e-03, -2.65635569e-02, -6.00279914e-03,  3.80281336e-03,
-3.54232267e-03,  1.36631634e-02, -7.93054327e-03, -5.16653853e-03,
-4.54091188e-03,  4.76645958e-03, -1.80741516e-03, -5.99573599e-03,
 1.96028012e-03, -1.68454312e-02, -1.20790526e-02, -2.52328953e-03,
-9.43484623e-03,  1.64717659e-02, -5.03119919e-03,  1.72983129e-02,
 6.79122796e-03, -1.59469731e-02, -5.26221236e-03, -7.80042494e-03,
 1.77786476e-03,  7.66355358e-03, -1.04484614e-02,  3.77084687e-03,
-1.98246222e-02, -1.45429904e-02, -3.40409018e-03, -6.80755824e-03,
-6.00427622e-03, -6.00387808e-03, -3.05296970e-03, -3.66698671e-03,
-1.95456445e-02, -6.00110600e-03, -1.49876392e-02, -3.55679658e-03,
 1.02266262e-03, -1.12012243e-02, -4.99977823e-03, -3.15652671e-03,
 1.00699312e-03, -4.00474761e-03,  1.06450506e-02,  1.35641862e-02,
-6.04864908e-03, -6.00297563e-03, -3.92426411e-03,  3.29488665e-02,
-1.89786274e-02,  4.58686030e-04, -6.00377843e-03,  1.23900604e-02,
-3.03337281e-03,  1.16271281e-03, -3.30965896e-03,  4.20221686e-03,
 2.34743338e-02,  2.43265764e-03, -3.42777255e-03, -3.35360790e-04,
-5.16253337e-02, -6.00461196e-03, -6.00307900e-03, -6.00464130e-03,
 1.43005722e-03,  4.59812162e-03,  1.84254218e-02, -3.34755145e-03,
-9.29790456e-03,  4.66804486e-03, -6.00451883e-03, -6.00398472e-03,
 1.37981717e-02, -3.87562928e-03, -4.08005854e-03, -6.00422686e-03,
-3.38513847e-03, -7.02523859e-03,  1.44284731e-02, -6.00450952e-03,
-1.57754794e-02, -3.61155812e-03, -2.95013399e-03,  1.12323631e-02,
-7.02035008e-03,  5.52526256e-03, -6.00365130e-03,  2.22522095e-02,
-1.43709769e-02, -3.54544236e-03,  6.80382270e-03,  2.69351900e-03,
 2.39149842e-04,  7.88299274e-03, -5.34355640e-03, -3.05824783e-02,
 1.97073584e-03, -1.13058528e-02, -5.65877371e-03, -6.00466924e-03,
-2.07305769e-03,  4.76630265e-03, -4.12163511e-03, -1.21525405e-02,
-7.71215092e-03, -4.54826001e-03,  1.06366212e-02, -4.65970077e-02,
 4.86294739e-04, -8.70221294e-03, -4.04643314e-03, -1.43065546e-02,
-6.00711443e-03,  8.58738180e-03, -1.01569633e-03, -7.09496532e-03,
-1.01917936e-02, -8.22837639e-04, -6.00444572e-03,  2.50869244e-03,
 1.36821289e-02, -2.51769181e-03, -4.07299353e-03, -4.32161381e-03,
 6.02753647e-03,  6.59000501e-03,  1.11173997e-02, -3.10064084e-03,
-4.45215451e-03, -3.98592092e-03, -6.00211322e-03,  7.65296165e-04,
-9.59617144e-04,  6.31386880e-03, -6.00285875e-03,  3.06496229e-02,
 2.15505203e-03, -5.99173363e-03,  8.29120912e-03, -8.36069975e-03,
-1.99382156e-02,  3.58867110e-03, -5.67910401e-03, -1.56062935e-02,
-6.00320613e-03, -5.14288712e-03, -7.78724672e-03, -1.26721403e-02,
 2.32999888e-03,  4.20920644e-03, -5.83444489e-03, -3.89796728e-03,
-3.33546568e-03,  1.06287356e-02,  3.96552449e-03, -6.00468274e-03,
-6.00014022e-03, -6.00353675e-03, -3.31809483e-02, -2.83884965e-02,
-2.96896906e-03, -1.96891418e-03, -9.86546092e-03,  1.01369098e-02,
```

```
        4.97551542e-03, -6.00377610e-03, -3.34706716e-03, -6.01534918e-03,
       -6.00211788e-03, -3.18017346e-03, -4.41927975e-03, -2.02464424e-02,
       -6.00385852e-03, -1.81741677e-02,  1.91682614e-02, -5.95156243e-03,
       -5.66643290e-03, -6.00378541e-03, -1.72791649e-02,  1.66675709e-02,
       -6.00388460e-03,  8.49588949e-04, -1.13889622e-02, -1.60178126e-04,
       -6.00072462e-03, -7.03945896e-03, -2.61450782e-02, -6.00253977e-03,
       -4.09829663e-03, -6.00339705e-03, -6.00322103e-03, -1.42162964e-02,
       -4.90964204e-03, -6.00360101e-03, -7.90582597e-03, -9.10534151e-03],
      dtype=float32),
array([[-0.0938881 ,  0.04190071,  0.02893946, -0.02560814,  0.03228481],
       [-0.1888359 ,  0.05349169,  0.01654298,  0.00796052,  0.08974966],
       [-0.119164  , -0.07860665,  0.03637654, -0.09344073,  0.08286934],
       ...,
       [ 0.03783495, -0.085783  ,  0.02580419,  0.03044481,  0.01621568],
       [-0.08817358, -0.08604153,  0.06678672,  0.02582594, -0.09324099],
       [ 0.0390219 , -0.086462  ,  0.0124651 , -0.03119957, -0.06505394]],
      dtype=float32),
array([-0.00424703, -0.02300429, -0.00714327,  0.00700427,  0.00833481],
      dtype=float32)]
```

Saving model

In [27]:
```
if os.path.isfile("cnn.h5") is False:
    cnn.save("cnn.h5")
```

Real time acc

In [28]:
```
def Predict(img):
    class_ = np.argmax(cnn.predict(img))
    return input_types[class_]
```

In [29]:
```
mp_drawing = mp.solutions.drawing_utils
mp_holistic = mp.solutions.holistic
```

In [30]:
```
holistic = mp_holistic.Holistic()
```

In [31]:

```
R = 25
thickness = 2


webcam = 0
capture = cv.VideoCapture(webcam)

fps = int(capture.get(cv.CAP_PROP_FPS))
print("fps is "+str(fps))

_, frame = capture.read()
height, width, channel = frame.shape

while capture.isOpened():
    #time.sleep()
    if cv.waitKey(1) & 0xFF == 13:
        break

    black = np.zeros(shape = frame.shape)

    _, frame = capture.read()

    frame_rgb = cv.cvtColor(frame, cv.COLOR_BGR2RGB)

    result = holistic.process(frame_rgb)

    try:
        hand_landmarks = result.right_hand_landmarks.landmark
        if hand_landmarks:
            x_max = 0
            y_max = 0
            x_min = width
            y_min = height
            for i in range(0,21,1):
                lm = hand_landmarks[i]
                x, y = int(lm.x * width), int(lm.y * height)
                if x > x_max:
                    x_max = x
                if x < x_min:
                    x_min = x
                if y > y_max:
```

```
                y_max = y
            if y < y_min:
                y_min = y

        frame_bgr = cv.cvtColor(frame_rgb, cv.COLOR_RGB2BGR)

        mp_drawing.draw_landmarks(frame_bgr, result.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS)

        cv.rectangle(frame_bgr, (x_min - R, y_min - R), (x_max + R, y_max + R), (0, 255, 0), thickness)

        result1 = frame_bgr

        mirror1 = cv.flip(result1, 1)


        '''

        '''
        mp_drawing.draw_landmarks(black, result.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS)

        croped = black[y_min - R + thickness: y_max + R - thickness, x_min - R + thickness : x_max + R - thickness
]

        resized = cv.resize(croped, (96, 96))

        mirror2 = cv.flip(resized, 1)

        result2 = mirror2

        img_mat = np.array([result2])

        class_ = Predict(img_mat)

        cv.putText(mirror1, str(class_), (100, 100), cv.FONT_HERSHEY_PLAIN, 2, (255,0,0), 1)

        cv.imshow("Frame2", result2)
    except:
        result1 = frame
        mirror1 = cv.flip(result1, 1)
        #result2 = black
        pass
```

```
    cv.imshow('frame1', mirror1)

capture.release()
cv.destroyAllWindows()
```

```
fps is 30
```

In [ ]: