

```
In [ ]: import numpy as np
import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from tensorflow.keras.optimizers import SGD,Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers import Dense,Activation,Dropout,LSTM,BatchNormalization
from keras.layers import Flatten
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.utils import to_categorical
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
```

```
In [ ]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
In [ ]: data_generator = ImageDataGenerator(rescale=1./255,validation_split=0.1,width_
shift_range=0.2,height_shift_range=0.2,zoom_range=0.2,horizontal_flip=True)
```

```
In [ ]: trainset='/content/drive/MyDrive/Colab Notebooks/Hands/Hands'
```

```
In [ ]: train_dataset=data_generator.flow_from_directory(trainset,
                                                    target_size=(100,100),
                                                    batch_size=128,
                                                    class_mode='categorical',
                                                    subset="training",
                                                    shuffle=True,)
validation_set=data_generator.flow_from_directory(trainset,
                                                    target_size=(100,100),
                                                    batch_size=128,
                                                    class_mode='categorical',
                                                    subset="validation",
                                                    shuffle=True)
```

Found 51 images belonging to 5 classes.

Found 2 images belonging to 5 classes.

```
In [ ]: validation_set.class_indices
```

```
Out[ ]: {'a': 0, 'b': 1, 'c': 2, 'd': 3, 'e': 4}
```

```
In [ ]: model = Sequential()
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding = 'same',input_shape=(100,100,3)))
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding = 'same'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding = 'same'))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding = 'same'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding = 'same'))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding = 'same'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(256,activation='relu',kernel_initializer='he_uniform'))
model.add(Dropout(0.2))
model.add(Dense(5,activation='softmax'))
```

```
In [ ]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])  
        model.fit(train_dataset,batch_size=128,epochs=100,verbose=1,validation_data=validation_set)
```

Epoch 1/100
1/1 [=====] - 5s 5s/step - loss: 1.7051 - accuracy:
0.2549 - val_loss: 24.4337 - val_accuracy: 0.0000e+00
Epoch 2/100
1/1 [=====] - 4s 4s/step - loss: 14.2951 - accuracy:
0.2353 - val_loss: 11.2092 - val_accuracy: 0.0000e+00
Epoch 3/100
1/1 [=====] - 4s 4s/step - loss: 7.8263 - accuracy:
0.1765 - val_loss: 2.4871 - val_accuracy: 0.5000
Epoch 4/100
1/1 [=====] - 4s 4s/step - loss: 2.3762 - accuracy:
0.2157 - val_loss: 1.0420 - val_accuracy: 0.5000
Epoch 5/100
1/1 [=====] - 4s 4s/step - loss: 1.6985 - accuracy:
0.2549 - val_loss: 1.2670 - val_accuracy: 0.5000
Epoch 6/100
1/1 [=====] - 4s 4s/step - loss: 1.5710 - accuracy:
0.3137 - val_loss: 1.4883 - val_accuracy: 0.5000
Epoch 7/100
1/1 [=====] - 4s 4s/step - loss: 1.5681 - accuracy:
0.3333 - val_loss: 1.3837 - val_accuracy: 0.5000
Epoch 8/100
1/1 [=====] - 4s 4s/step - loss: 1.5436 - accuracy:
0.3137 - val_loss: 1.1910 - val_accuracy: 0.5000
Epoch 9/100
1/1 [=====] - 4s 4s/step - loss: 1.5873 - accuracy:
0.3137 - val_loss: 1.2382 - val_accuracy: 0.5000
Epoch 10/100
1/1 [=====] - 4s 4s/step - loss: 1.5142 - accuracy:
0.3333 - val_loss: 1.2646 - val_accuracy: 0.5000
Epoch 11/100
1/1 [=====] - 4s 4s/step - loss: 1.5067 - accuracy:
0.3529 - val_loss: 1.2069 - val_accuracy: 0.5000
Epoch 12/100
1/1 [=====] - 4s 4s/step - loss: 1.4878 - accuracy:
0.3137 - val_loss: 1.1347 - val_accuracy: 0.5000
Epoch 13/100
1/1 [=====] - 4s 4s/step - loss: 1.4669 - accuracy:
0.3137 - val_loss: 1.3296 - val_accuracy: 0.5000
Epoch 14/100
1/1 [=====] - 4s 4s/step - loss: 1.4305 - accuracy:
0.3333 - val_loss: 1.2052 - val_accuracy: 0.5000
Epoch 15/100
1/1 [=====] - 4s 4s/step - loss: 1.3414 - accuracy:
0.3529 - val_loss: 0.9428 - val_accuracy: 1.0000
Epoch 16/100
1/1 [=====] - 4s 4s/step - loss: 1.2353 - accuracy:
0.4118 - val_loss: 0.9989 - val_accuracy: 0.5000
Epoch 17/100
1/1 [=====] - 4s 4s/step - loss: 1.2276 - accuracy:
0.3529 - val_loss: 0.9953 - val_accuracy: 1.0000
Epoch 18/100
1/1 [=====] - 4s 4s/step - loss: 1.2554 - accuracy:
0.6078 - val_loss: 0.7171 - val_accuracy: 0.5000
Epoch 19/100
1/1 [=====] - 4s 4s/step - loss: 1.1181 - accuracy:
0.3529 - val_loss: 0.9591 - val_accuracy: 1.0000

Epoch 20/100
1/1 [=====] - 4s 4s/step - loss: 0.9991 - accuracy:
0.7059 - val_loss: 0.5301 - val_accuracy: 1.0000
Epoch 21/100
1/1 [=====] - 4s 4s/step - loss: 0.9001 - accuracy:
0.6275 - val_loss: 0.4678 - val_accuracy: 1.0000
Epoch 22/100
1/1 [=====] - 4s 4s/step - loss: 0.8754 - accuracy:
0.6471 - val_loss: 1.1719 - val_accuracy: 0.0000e+00
Epoch 23/100
1/1 [=====] - 4s 4s/step - loss: 0.9339 - accuracy:
0.6667 - val_loss: 0.0444 - val_accuracy: 1.0000
Epoch 24/100
1/1 [=====] - 4s 4s/step - loss: 1.0903 - accuracy:
0.4706 - val_loss: 1.4876 - val_accuracy: 0.5000
Epoch 25/100
1/1 [=====] - 4s 4s/step - loss: 1.1650 - accuracy:
0.4902 - val_loss: 1.3962 - val_accuracy: 0.5000
Epoch 26/100
1/1 [=====] - 4s 4s/step - loss: 1.0792 - accuracy:
0.4902 - val_loss: 0.9085 - val_accuracy: 0.5000
Epoch 27/100
1/1 [=====] - 4s 4s/step - loss: 0.8673 - accuracy:
0.6275 - val_loss: 0.3710 - val_accuracy: 1.0000
Epoch 28/100
1/1 [=====] - 4s 4s/step - loss: 0.9948 - accuracy:
0.5294 - val_loss: 1.0754 - val_accuracy: 0.5000
Epoch 29/100
1/1 [=====] - 4s 4s/step - loss: 0.8007 - accuracy:
0.6667 - val_loss: 1.0862 - val_accuracy: 0.0000e+00
Epoch 30/100
1/1 [=====] - 4s 4s/step - loss: 0.8956 - accuracy:
0.6275 - val_loss: 0.5945 - val_accuracy: 1.0000
Epoch 31/100
1/1 [=====] - 4s 4s/step - loss: 0.8975 - accuracy:
0.5294 - val_loss: 0.2484 - val_accuracy: 1.0000
Epoch 32/100
1/1 [=====] - 4s 4s/step - loss: 0.7249 - accuracy:
0.6471 - val_loss: 0.1177 - val_accuracy: 1.0000
Epoch 33/100
1/1 [=====] - 4s 4s/step - loss: 0.9613 - accuracy:
0.5098 - val_loss: 0.4582 - val_accuracy: 1.0000
Epoch 34/100
1/1 [=====] - 4s 4s/step - loss: 0.6695 - accuracy:
0.7647 - val_loss: 0.7289 - val_accuracy: 0.5000
Epoch 35/100
1/1 [=====] - 4s 4s/step - loss: 0.7504 - accuracy:
0.7059 - val_loss: 1.1443 - val_accuracy: 0.5000
Epoch 36/100
1/1 [=====] - 4s 4s/step - loss: 0.6732 - accuracy:
0.7255 - val_loss: 0.5968 - val_accuracy: 0.5000
Epoch 37/100
1/1 [=====] - 4s 4s/step - loss: 0.6755 - accuracy:
0.7451 - val_loss: 0.1972 - val_accuracy: 1.0000
Epoch 38/100
1/1 [=====] - 4s 4s/step - loss: 0.6414 - accuracy:
0.7647 - val_loss: 0.3245 - val_accuracy: 1.0000

Epoch 39/100
1/1 [=====] - 4s 4s/step - loss: 0.4308 - accuracy: 0.8431 - val_loss: 0.2658 - val_accuracy: 1.0000
Epoch 40/100
1/1 [=====] - 4s 4s/step - loss: 0.4631 - accuracy: 0.7843 - val_loss: 0.6109 - val_accuracy: 0.5000
Epoch 41/100
1/1 [=====] - 4s 4s/step - loss: 0.5939 - accuracy: 0.7059 - val_loss: 0.7820 - val_accuracy: 0.5000
Epoch 42/100
1/1 [=====] - 4s 4s/step - loss: 0.3774 - accuracy: 0.8627 - val_loss: 0.2281 - val_accuracy: 1.0000
Epoch 43/100
1/1 [=====] - 4s 4s/step - loss: 0.7486 - accuracy: 0.7059 - val_loss: 1.5370 - val_accuracy: 0.5000
Epoch 44/100
1/1 [=====] - 4s 4s/step - loss: 0.5421 - accuracy: 0.7451 - val_loss: 1.0389 - val_accuracy: 0.5000
Epoch 45/100
1/1 [=====] - 4s 4s/step - loss: 0.4882 - accuracy: 0.7647 - val_loss: 0.0804 - val_accuracy: 1.0000
Epoch 46/100
1/1 [=====] - 4s 4s/step - loss: 0.6323 - accuracy: 0.7647 - val_loss: 1.0576 - val_accuracy: 0.5000
Epoch 47/100
1/1 [=====] - 4s 4s/step - loss: 0.4250 - accuracy: 0.8824 - val_loss: 0.9816 - val_accuracy: 0.0000e+00
Epoch 48/100
1/1 [=====] - 4s 4s/step - loss: 0.4648 - accuracy: 0.8039 - val_loss: 1.2206 - val_accuracy: 0.5000
Epoch 49/100
1/1 [=====] - 4s 4s/step - loss: 0.4938 - accuracy: 0.8431 - val_loss: 0.7362 - val_accuracy: 0.5000
Epoch 50/100
1/1 [=====] - 4s 4s/step - loss: 0.3447 - accuracy: 0.8824 - val_loss: 0.3738 - val_accuracy: 0.5000
Epoch 51/100
1/1 [=====] - 4s 4s/step - loss: 0.5700 - accuracy: 0.7647 - val_loss: 0.4404 - val_accuracy: 1.0000
Epoch 52/100
1/1 [=====] - 4s 4s/step - loss: 0.4227 - accuracy: 0.8039 - val_loss: 0.9907 - val_accuracy: 0.5000
Epoch 53/100
1/1 [=====] - 4s 4s/step - loss: 0.4454 - accuracy: 0.7843 - val_loss: 0.7164 - val_accuracy: 0.5000
Epoch 54/100
1/1 [=====] - 4s 4s/step - loss: 0.4937 - accuracy: 0.8431 - val_loss: 0.3327 - val_accuracy: 1.0000
Epoch 55/100
1/1 [=====] - 4s 4s/step - loss: 0.3620 - accuracy: 0.8235 - val_loss: 0.3867 - val_accuracy: 1.0000
Epoch 56/100
1/1 [=====] - 4s 4s/step - loss: 0.4070 - accuracy: 0.8431 - val_loss: 0.1721 - val_accuracy: 1.0000
Epoch 57/100
1/1 [=====] - 4s 4s/step - loss: 0.3733 - accuracy: 0.8824 - val_loss: 0.7266 - val_accuracy: 0.5000

Epoch 58/100
1/1 [=====] - 4s 4s/step - loss: 0.3875 - accuracy:
0.8235 - val_loss: 0.6343 - val_accuracy: 0.5000
Epoch 59/100
1/1 [=====] - 4s 4s/step - loss: 0.4613 - accuracy:
0.8039 - val_loss: 0.5094 - val_accuracy: 0.5000
Epoch 60/100
1/1 [=====] - 4s 4s/step - loss: 0.3112 - accuracy:
0.8039 - val_loss: 0.5837 - val_accuracy: 0.5000
Epoch 61/100
1/1 [=====] - 4s 4s/step - loss: 0.1990 - accuracy:
0.9216 - val_loss: 0.2806 - val_accuracy: 1.0000
Epoch 62/100
1/1 [=====] - 4s 4s/step - loss: 0.5333 - accuracy:
0.7843 - val_loss: 0.8425 - val_accuracy: 0.5000
Epoch 63/100
1/1 [=====] - 5s 5s/step - loss: 0.2948 - accuracy:
0.8627 - val_loss: 1.5089 - val_accuracy: 0.0000e+00
Epoch 64/100
1/1 [=====] - 4s 4s/step - loss: 0.2192 - accuracy:
0.9020 - val_loss: 0.7725 - val_accuracy: 0.5000
Epoch 65/100
1/1 [=====] - 4s 4s/step - loss: 0.3197 - accuracy:
0.8824 - val_loss: 0.6822 - val_accuracy: 0.5000
Epoch 66/100
1/1 [=====] - 4s 4s/step - loss: 0.2294 - accuracy:
0.9216 - val_loss: 0.6702 - val_accuracy: 0.5000
Epoch 67/100
1/1 [=====] - 4s 4s/step - loss: 0.2670 - accuracy:
0.9020 - val_loss: 0.3184 - val_accuracy: 1.0000
Epoch 68/100
1/1 [=====] - 4s 4s/step - loss: 0.4015 - accuracy:
0.8627 - val_loss: 0.3889 - val_accuracy: 1.0000
Epoch 69/100
1/1 [=====] - 4s 4s/step - loss: 0.2852 - accuracy:
0.8824 - val_loss: 0.6100 - val_accuracy: 0.5000
Epoch 70/100
1/1 [=====] - 4s 4s/step - loss: 0.2634 - accuracy:
0.9020 - val_loss: 3.8816 - val_accuracy: 0.5000
Epoch 71/100
1/1 [=====] - 4s 4s/step - loss: 0.2153 - accuracy:
0.9020 - val_loss: 0.0565 - val_accuracy: 1.0000
Epoch 72/100
1/1 [=====] - 4s 4s/step - loss: 0.2622 - accuracy:
0.9216 - val_loss: 0.1479 - val_accuracy: 1.0000
Epoch 73/100
1/1 [=====] - 4s 4s/step - loss: 0.2943 - accuracy:
0.9020 - val_loss: 0.6771 - val_accuracy: 0.5000
Epoch 74/100
1/1 [=====] - 4s 4s/step - loss: 0.4438 - accuracy:
0.9020 - val_loss: 0.1382 - val_accuracy: 1.0000
Epoch 75/100
1/1 [=====] - 4s 4s/step - loss: 0.3007 - accuracy:
0.8824 - val_loss: 0.5246 - val_accuracy: 0.5000
Epoch 76/100
1/1 [=====] - 4s 4s/step - loss: 0.2963 - accuracy:
0.8824 - val_loss: 0.8812 - val_accuracy: 0.5000

Epoch 77/100
1/1 [=====] - 4s 4s/step - loss: 0.3225 - accuracy:
0.8627 - val_loss: 1.5606 - val_accuracy: 0.5000
Epoch 78/100
1/1 [=====] - 4s 4s/step - loss: 0.2727 - accuracy:
0.8824 - val_loss: 0.0385 - val_accuracy: 1.0000
Epoch 79/100
1/1 [=====] - 4s 4s/step - loss: 0.3199 - accuracy:
0.8431 - val_loss: 0.3593 - val_accuracy: 1.0000
Epoch 80/100
1/1 [=====] - 4s 4s/step - loss: 0.3155 - accuracy:
0.8627 - val_loss: 1.0654 - val_accuracy: 0.0000e+00
Epoch 81/100
1/1 [=====] - 4s 4s/step - loss: 0.1986 - accuracy:
0.9412 - val_loss: 0.1659 - val_accuracy: 1.0000
Epoch 82/100
1/1 [=====] - 4s 4s/step - loss: 0.1744 - accuracy:
0.9412 - val_loss: 0.1583 - val_accuracy: 1.0000
Epoch 83/100
1/1 [=====] - 4s 4s/step - loss: 0.1425 - accuracy:
0.9412 - val_loss: 0.2935 - val_accuracy: 1.0000
Epoch 84/100
1/1 [=====] - 5s 5s/step - loss: 0.2234 - accuracy:
0.9020 - val_loss: 0.4008 - val_accuracy: 0.5000
Epoch 85/100
1/1 [=====] - 4s 4s/step - loss: 0.1539 - accuracy:
0.9412 - val_loss: 0.8640 - val_accuracy: 0.5000
Epoch 86/100
1/1 [=====] - 4s 4s/step - loss: 0.2039 - accuracy:
0.9020 - val_loss: 0.3237 - val_accuracy: 1.0000
Epoch 87/100
1/1 [=====] - 4s 4s/step - loss: 0.1577 - accuracy:
0.9412 - val_loss: 0.3988 - val_accuracy: 0.5000
Epoch 88/100
1/1 [=====] - 4s 4s/step - loss: 0.1482 - accuracy:
0.9020 - val_loss: 0.7331 - val_accuracy: 0.5000
Epoch 89/100
1/1 [=====] - 4s 4s/step - loss: 0.0852 - accuracy:
0.9804 - val_loss: 0.9625 - val_accuracy: 0.5000
Epoch 90/100
1/1 [=====] - 4s 4s/step - loss: 0.2628 - accuracy:
0.9216 - val_loss: 1.5774 - val_accuracy: 0.5000
Epoch 91/100
1/1 [=====] - 4s 4s/step - loss: 0.2116 - accuracy:
0.9216 - val_loss: 0.3679 - val_accuracy: 1.0000
Epoch 92/100
1/1 [=====] - 4s 4s/step - loss: 0.2925 - accuracy:
0.8824 - val_loss: 0.0724 - val_accuracy: 1.0000
Epoch 93/100
1/1 [=====] - 4s 4s/step - loss: 0.1653 - accuracy:
0.9216 - val_loss: 0.0168 - val_accuracy: 1.0000
Epoch 94/100
1/1 [=====] - 4s 4s/step - loss: 0.1810 - accuracy:
0.9412 - val_loss: 0.0709 - val_accuracy: 1.0000
Epoch 95/100
1/1 [=====] - 4s 4s/step - loss: 0.1015 - accuracy:
0.9804 - val_loss: 0.4357 - val_accuracy: 0.5000


```

Epoch 96/100
1/1 [=====] - 4s 4s/step - loss: 0.1096 - accuracy:
0.9608 - val_loss: 0.0483 - val_accuracy: 1.0000
Epoch 97/100
1/1 [=====] - 4s 4s/step - loss: 0.1346 - accuracy:
0.9412 - val_loss: 0.4697 - val_accuracy: 0.5000
Epoch 98/100
1/1 [=====] - 4s 4s/step - loss: 0.1688 - accuracy:
0.9412 - val_loss: 0.7116 - val_accuracy: 0.5000
Epoch 99/100
1/1 [=====] - 4s 4s/step - loss: 0.1024 - accuracy:
0.9608 - val_loss: 0.7660 - val_accuracy: 0.5000
Epoch 100/100
1/1 [=====] - 4s 4s/step - loss: 0.0768 - accuracy:
0.9804 - val_loss: 0.4091 - val_accuracy: 0.5000

```

Out[]: <keras.callbacks.History at 0x7fb53cbb5c90>

```

In [ ]: train_dataset=data_generator.flow_from_directory(trainset,
                                                    target_size=(100,100),
                                                    batch_size=128,
                                                    class_mode='categorical',
                                                    subset="training",
                                                    shuffle=True,)
validation_set=data_generator.flow_from_directory(trainset,
                                                    target_size=(100,100),
                                                    batch_size=128,
                                                    class_mode='categorical',
                                                    subset="validation",
                                                    shuffle=True)

```

Found 51 images belonging to 5 classes.
Found 2 images belonging to 5 classes.

```
In [ ]: model.save('chitay.hs')
```

INFO:tensorflow:Assets written to: chitay.hs/assets

```
In [ ]: model1=load_model('chitay.hs')
```

```

In [ ]: import numpy as np
from tensorflow.keras.utils import load_img, img_to_array
img=load_img('Hand_0003133.jpg', target_size=(100,100))
img=img_to_array(img)
img=img.reshape(1,100,100,3)
img=img.astype('float32')
img=img/255
np.argmax(model1.predict(img), axis=-1)

```

Out[]: array([1])

```
In [ ]: label = ['a', 'b', 'c', 'd', 'e']
```

```
In [ ]: import numpy as np
from tensorflow.keras.utils import load_img, img_to_array
img = load_img('Hand_0003133.jpg', target_size = (100,100))
plt.imshow(img)
img = img_to_array(img)
img = img.reshape(1,100,100,3)
img = img.astype('float32')
img = img/255
print(label[np.argmax(model.predict(img))])
```

b

