

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import numpy as np
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
```

```
training_set = tf.keras.utils.image_dataset_from_directory(
    '/content/drive/MyDrive/train',
    labels="inferred",
    label_mode="categorical",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(64, 64),
    shuffle=True,
    seed=None,
    validation_split=None,
    subset=None,
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False,
)
```

Found 3115 files belonging to 36 classes.

```
validation_set = tf.keras.utils.image_dataset_from_directory(
    '/content/drive/MyDrive/validation',
    labels="inferred",
    label_mode="categorical",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(64, 64),
    shuffle=True,
    seed=None,
    validation_split=None,
    subset=None,
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False,
)
```

Found 351 files belonging to 36 classes.

```
cnn= tf.keras.models.Sequential()
```

```
cnn.add(tf.keras.layers.Conv2D(filters= 64, kernel_size=3, activation= 'relu', input_shape= [64,64,3] ))
cnn.add(tf.keras.layers.MaxPooling2D(pool_size=2, strides=2,))
```

```
cnn.add(tf.keras.layers.Conv2D(filters= 64, kernel_size=3, activation= 'relu'))
cnn.add(tf.keras.layers.MaxPooling2D(pool_size=2, strides=2,))
```

```
cnn.add(tf.keras.layers.Dropout(0.5))
```

```
cnn.add(tf.keras.layers.Flatten())
```

```
cnn.add(tf.keras.layers.Dense(units= 128,activation='relu' ))
```

```
cnn.add(tf.keras.layers.Dense(units= 36, activation= 'softmax'))
```

```
cnn.compile(optimizer= 'rmsprop', loss= 'categorical_crossentropy', metrics= ['accuracy'])
```

```
cnn.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 62, 62, 64)	1792
max_pooling2d_1 (MaxPooling2D)	(None, 31, 31, 64)	0
conv2d_6 (Conv2D)	(None, 29, 29, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_1 (Dropout)	(None, 14, 14, 64)	0
flatten_1 (Flatten)	(None, 12544)	0
dense (Dense)	(None, 128)	1605760
dense_1 (Dense)	(None, 36)	4644

=====
Total params: 1649124 (6.29 MB)
Trainable params: 1649124 (6.29 MB)
Non-trainable params: 0 (0.00 Byte)

cnn.fit(x= training_set, validation_data= validation_set, epochs= 40)

```
Epoch 1/40
98/98 [=====] - 595s 6s/step - loss: 10.4643 - accuracy: 0.0289 - val_loss: 3.5151 - val_accuracy: 0.0712
Epoch 2/40
98/98 [=====] - 72s 681ms/step - loss: 4.8975 - accuracy: 0.0478 - val_loss: 3.3079 - val_accuracy: 0.1197
Epoch 3/40
98/98 [=====] - 76s 736ms/step - loss: 3.7018 - accuracy: 0.0783 - val_loss: 3.2128 - val_accuracy: 0.1652
Epoch 4/40
98/98 [=====] - 73s 694ms/step - loss: 3.7399 - accuracy: 0.0950 - val_loss: 3.0021 - val_accuracy: 0.2393
Epoch 5/40
98/98 [=====] - 72s 689ms/step - loss: 3.4989 - accuracy: 0.1515 - val_loss: 3.2403 - val_accuracy: 0.2165
Epoch 6/40
98/98 [=====] - 72s 700ms/step - loss: 3.4666 - accuracy: 0.2106 - val_loss: 2.5589 - val_accuracy: 0.3875
Epoch 7/40
98/98 [=====] - 75s 726ms/step - loss: 3.0967 - accuracy: 0.2520 - val_loss: 2.5550 - val_accuracy: 0.4330
Epoch 8/40
98/98 [=====] - 74s 705ms/step - loss: 3.5369 - accuracy: 0.2912 - val_loss: 1.8939 - val_accuracy: 0.5328
Epoch 9/40
98/98 [=====] - 74s 710ms/step - loss: 3.0106 - accuracy: 0.3486 - val_loss: 6.4807 - val_accuracy: 0.2678
Epoch 10/40
98/98 [=====] - 73s 698ms/step - loss: 2.5539 - accuracy: 0.4071 - val_loss: 1.8520 - val_accuracy: 0.5783
Epoch 11/40
98/98 [=====] - 81s 791ms/step - loss: 2.8792 - accuracy: 0.4279 - val_loss: 1.6175 - val_accuracy: 0.6610
Epoch 12/40
98/98 [=====] - 74s 708ms/step - loss: 2.7113 - accuracy: 0.4671 - val_loss: 1.4182 - val_accuracy: 0.7066
Epoch 13/40
98/98 [=====] - 75s 726ms/step - loss: 2.4550 - accuracy: 0.5133 - val_loss: 1.6582 - val_accuracy: 0.6439
Epoch 14/40
98/98 [=====] - 72s 698ms/step - loss: 2.0401 - accuracy: 0.5352 - val_loss: 2.2947 - val_accuracy: 0.5755
Epoch 15/40
98/98 [=====] - 74s 711ms/step - loss: 1.8903 - accuracy: 0.5637 - val_loss: 1.5570 - val_accuracy: 0.6923
Epoch 16/40
98/98 [=====] - 79s 756ms/step - loss: 1.6173 - accuracy: 0.6148 - val_loss: 1.2029 - val_accuracy: 0.7835
Epoch 17/40
98/98 [=====] - 76s 737ms/step - loss: 1.9085 - accuracy: 0.6186 - val_loss: 3.0418 - val_accuracy: 0.5385
Epoch 18/40
98/98 [=====] - 76s 731ms/step - loss: 1.5388 - accuracy: 0.6584 - val_loss: 1.4193 - val_accuracy: 0.8348
Epoch 19/40
98/98 [=====] - 80s 745ms/step - loss: 2.3000 - accuracy: 0.6600 - val_loss: 2.0354 - val_accuracy: 0.7920
Epoch 20/40
98/98 [=====] - 76s 728ms/step - loss: 1.8946 - accuracy: 0.6735 - val_loss: 3.9052 - val_accuracy: 0.5527
Epoch 21/40
98/98 [=====] - 76s 731ms/step - loss: 1.5989 - accuracy: 0.7037 - val_loss: 1.3158 - val_accuracy: 0.8547
Epoch 22/40
98/98 [=====] - 76s 735ms/step - loss: 1.3701 - accuracy: 0.7140 - val_loss: 1.1342 - val_accuracy: 0.8575
Epoch 23/40
98/98 [=====] - 75s 713ms/step - loss: 1.3471 - accuracy: 0.7242 - val_loss: 1.4722 - val_accuracy: 0.8462
Epoch 24/40
98/98 [=====] - 76s 728ms/step - loss: 1.1907 - accuracy: 0.7451 - val_loss: 1.7304 - val_accuracy: 0.8234
Epoch 25/40
98/98 [=====] - 78s 752ms/step - loss: 1.3121 - accuracy: 0.7432 - val_loss: 2.0621 - val_accuracy: 0.7550
Epoch 26/40
98/98 [=====] - 83s 801ms/step - loss: 1.3524 - accuracy: 0.7483 - val_loss: 2.7317 - val_accuracy: 0.7892
Epoch 27/40
```

```
98/98 [=====] - 76s 727ms/step - loss: 1.0941 - accuracy: 0.7884 - val_loss: 1.5627 - val_accuracy: 0.8348
Epoch 28/40
98/98 [=====] - 76s 738ms/step - loss: 1.0705 - accuracy: 0.7849 - val_loss: 1.1871 - val_accuracy: 0.8946
Epoch 29/40
98/98 [=====] - 77s 743ms/step - loss: 0.8975 - accuracy: 0.8013 - val_loss: 1.4618 - val_accuracy: 0.8632
```

```
cnn.save('trained_model.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `m
saving_api.save_model(
```

```
training_history = cnn.history
```

```
training_history.history
```

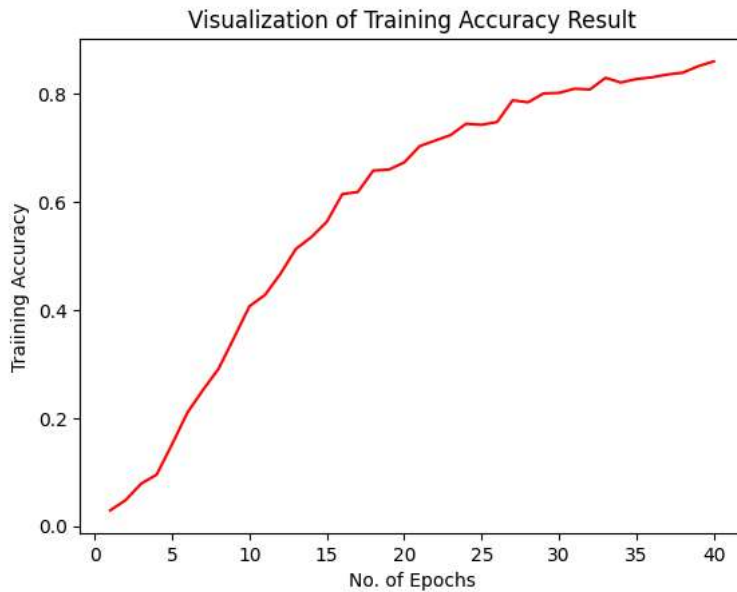
```
{'loss': [10.4642915725708,
4.897549152374268,
3.7017977237701416,
3.739884853363037,
3.4989089965820312,
3.4665539264678955,
3.0966737270355225,
3.5368833541870117,
3.010610818862915,
2.5538508892059326,
2.8792483806610107,
2.711282968521118,
2.4550259113311768,
2.0400776863098145,
1.89029860496521,
1.6172666549682617,
1.908538579940796,
1.538784384727478,
2.3000025749206543,
1.8945560455322266,
1.5989105701446533,
1.3700799942016602,
1.3470869064331055,
1.1906530857086182,
1.312118649482727,
1.352351427078247,
1.0940606594085693,
1.070489525794983,
0.897507905960083,
1.0882915258407593,
1.020885705947876,
1.157928228378296,
0.9852746725082397,
1.211165189743042,
0.9784979224205017,
0.9932981133460999,
1.2051795721054077,
0.997975766658783,
0.7895012497901917,
0.8991108536720276],
'accuracy': [0.028892455622553825,
0.04783306643366814,
0.07833065837621689,
0.09502407908439636,
0.1515248864889145,
0.21059389412403107,
0.2520064115524292,
0.29117175936698914,
0.34863564372062683,
0.4070625901222229,
0.4279293715953827,
0.46709468960762024,
0.513322651386261,
0.5351524949073792,
0.5637239217758179,
0.6147672533988953,
0.6186195611953735,
0.6584269404411316,
```

```
import json
with open('training_hist.json','w') as f:
    json.dump(training_history.history,f)
```

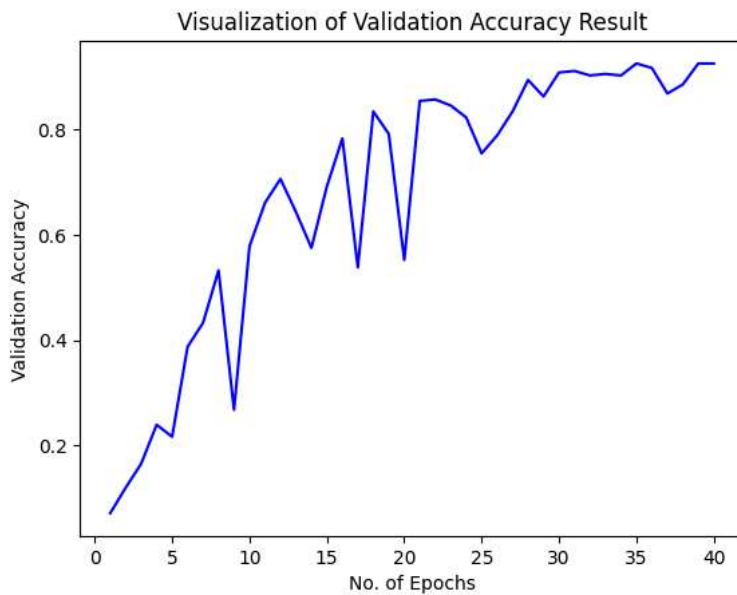
```
print("Validation set Accuracy: {} {}".format(training_history.history['val_accuracy'][-1]*100))
```

Validation set Accuracy: 92.59259104728699 %

```
epochs = [i for i in range(1,41)]
plt.plot(epochs,training_history.history['accuracy'],color='red')
plt.xlabel('No. of Epochs')
plt.ylabel('Training Accuracy')
plt.title('Visualization of Training Accuracy Result')
plt.show()
```



```
plt.plot(epochs,training_history.history['val_accuracy'],color='blue')
plt.xlabel('No. of Epochs')
plt.ylabel('Validation Accuracy')
plt.title('Visualization of Validation Accuracy Result')
plt.show()
```



-TEST SET EVALUATION

```
test_set = tf.keras.utils.image_dataset_from_directory(
    '/content/drive/MyDrive/test',
    labels="inferred",
    label_mode="categorical",
    ..
```

```
class_names=None,  
color_mode="rgb",  
batch_size=32,  
image_size=(64, 64),  
shuffle=True,  
seed=None,  
validation_split=None,  
subset=None,  
interpolation="bilinear",  
follow_links=False,  
crop_to_aspect_ratio=False  
)
```

Found 359 files belonging to 36 classes.

```
test_loss, test_acc = cnn.evaluate(test_set)  
print('Test accuracy:', test_acc)
```

```
12/12 [=====] - 110s 878ms/step - loss: 1.6272 - accuracy: 0.9248  
Test accuracy: 0.9247910976409912
```