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

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=9473189

Enter your authorization code:
··········
Mounted at /content/drive/
```

New Section

```
!pip install -q keras
```

```
import os, shutil
import numpy as np
import glob
import random
```

```
base_dir='/content/drive/My Drive/CMS_Solutions/data'
```

```
train_dir=os.path.join(base_dir,'train')
validation_dir=os.path.join(base_dir,'validation')
test_dir=os.path.join(base_dir,'test')
```

```
from keras import layers
from keras import models
model=models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation='relu',
input_shape=(200,200,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Flatten())
model.add(layers.Dense(512,activation='relu'))
model.add(layers.Dense(101,activation='softmax'))
```

```
from keras import optimizers
model.compile(loss='categorical_crossentropy',
optimizer=optimizers.RMSprop(lr=1e-4),
metrics=['acc'])
```

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255)
test_datagen=ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(train_dir,target_size=(200, 200),batch_si
```

```
Found 1837 images belonging to 102 classes.
```

```
datagen=ImageDataGenerator(
rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest')
```

```
model=models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation='relu',
input_shape=(200,200,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(512,activation='relu'))
model.add(layers.Dense(102,activation='softmax'))
model.compile(loss='categorical_crossentropy',
optimizer=optimizers.RMSprop(lr=1e-4),
metrics=['acc'])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
```

```
from keras import layers
from keras import models
model=models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation='relu',
input_shape=(200,200,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
```

```
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Flatten())
model.add(layers.Dense(512,activation='relu'))
model.add(layers.Dense(102,activation='softmax'))
```

Double-click (or enter) to edit

Double-click (or enter) to edit

```
from keras import optimizers
model.compile(loss='categorical_crossentropy',
optimizer=optimizers.RMSprop(lr=1e-4),
metrics=['acc'])
```

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255)
test_datagen=ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(train_dir,target_size=(200, 200),batch_si
```

```
Found 1837 images belonging to 102 classes.
```

```
validation_generator=test_datagen.flow_from_directory(validation_dir,target_size=(200,200),ba
```

```
Found 1020 images belonging to 102 classes.
```

```
from keras.applications import VGG16
conv_base=VGG16(weights='imagenet',
include_top=False,
input_shape=(200,200,3))
```

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download
58892288/58889256 [==============================] - 2s 0us/step
```

```
from keras import models
from keras import layers
model=models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256,activation='relu'))
model.add(layers.Dense(102,activation='softmax'))
```

```
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 6, 6, 512)         14714688
_________________________________________________________________
flatten_6 (Flatten)          (None, 18432)             0
_________________________________________________________________
dense_11 (Dense)             (None, 256)               4718848
_________________________________________________________________
dense_12 (Dense)             (None, 102)               26214
=================================================================
Total params: 19,459,750
Trainable params: 19,459,750
Non-trainable params: 0
_________________________________________________________________
```

```
from keras.preprocessing.image import ImageDataGenerator
from keras import optimizers
train_datagen=ImageDataGenerator(
rescale=1./255,
rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest')
test_datagen=ImageDataGenerator(rescale=1./255)
train_generator=train_datagen.flow_from_directory(
train_dir,
target_size=(200,200),
batch_size=17,
class_mode='categorical')
validation_generator=test_datagen.flow_from_directory(
validation_dir,
target_size=(200,200),
batch_size=17,
class_mode='categorical')
model.compile(loss='categorical_crossentropy',
optimizer=optimizers.RMSprop(lr=2e-5),
metrics=['acc'])
history=model.fit_generator(
train_generator,
steps_per_epoch=100,
epochs=30,
validation_data=validation_generator,
validation_steps=17)
```

```
Found 1837 images belonging to 102 classes.
Found 1020 images belonging to 102 classes.
Epoch 1/30
100/100 [==============================] - 61s 607ms/step - loss: 4.5768 - acc: 0.0335 -
Epoch 2/30
100/100 [==============================] - 55s 551ms/step - loss: 3.9943 - acc: 0.1488 -
Epoch 3/30
100/100 [==============================] - 55s 551ms/step - loss: 3.2181 - acc: 0.2647 -
Epoch 4/30
100/100 [==============================] - 55s 550ms/step - loss: 2.7684 - acc: 0.3517 -
Epoch 5/30
100/100 [==============================] - 55s 552ms/step - loss: 2.2526 - acc: 0.4535 -
Epoch 6/30
100/100 [==============================] - 55s 552ms/step - loss: 1.9729 - acc: 0.4971 -
Epoch 7/30
100/100 [==============================] - 55s 552ms/step - loss: 1.6838 - acc: 0.5682 -
Epoch 8/30
100/100 [==============================] - 56s 557ms/step - loss: 1.3887 - acc: 0.6276 -
Epoch 9/30
100/100 [==============================] - 55s 548ms/step - loss: 1.2386 - acc: 0.6641 -
Epoch 10/30
100/100 [==============================] - 56s 556ms/step - loss: 1.0726 - acc: 0.7100 -
Epoch 11/30
100/100 [==============================] - 55s 552ms/step - loss: 0.9804 - acc: 0.7336 -
Epoch 12/30
100/100 [==============================] - 55s 552ms/step - loss: 0.7975 - acc: 0.7701 -
Epoch 13/30
100/100 [==============================] - 55s 551ms/step - loss: 0.7770 - acc: 0.7771 -
Epoch 14/30
100/100 [==============================] - 55s 552ms/step - loss: 0.6707 - acc: 0.8077 -
Epoch 15/30
100/100 [==============================] - 55s 551ms/step - loss: 0.6131 - acc: 0.8247 -
Epoch 16/30
100/100 [==============================] - 55s 551ms/step - loss: 0.5721 - acc: 0.8323 -
Epoch 17/30
100/100 [==============================] - 55s 551ms/step - loss: 0.5008 - acc: 0.8565 -
Epoch 18/30
100/100 [==============================] - 55s 555ms/step - loss: 0.4780 - acc: 0.8676 -
Epoch 19/30
100/100 [==============================] - 55s 552ms/step - loss: 0.4349 - acc: 0.8812 -
Epoch 20/30
100/100 [==============================] - 55s 550ms/step - loss: 0.3914 - acc: 0.8806 -
Epoch 21/30
100/100 [==============================] - 55s 546ms/step - loss: 0.3788 - acc: 0.8917 -
Epoch 22/30
100/100 [==============================] - 55s 551ms/step - loss: 0.3265 - acc: 0.9106 -
Epoch 23/30
100/100 [==============================] - 55s 551ms/step - loss: 0.3709 - acc: 0.8959 -
Epoch 24/30
100/100 [==============================] - 55s 554ms/step - loss: 0.2930 - acc: 0.9147 -
Epoch 25/30
100/100 [==============================] - 55s 550ms/step - loss: 0.2568 - acc: 0.9229 -
Epoch 26/30
100/100 [==============================] - 55s 551ms/step - loss: 0.2886 - acc: 0.9182 -
Epoch 27/30
100/100 [==============================] - 55s 550ms/step - loss: 0.2421 - acc: 0.9282 -
Epoch 28/30
```

```
100/100 [==============================] - 55s 550ms/step - loss: 0.2452 - acc: 0.9212 -
Epoch 29/30
100/100 [==============================] - 55s 551ms/step - loss: 0.2257 - acc: 0.9359 -
```

```
import matplotlib.pyplot as plt
acc=history.history['acc']
val_acc=history.history['val_acc']
loss=history.history['loss']
val_loss=history.history['val_loss']
epochs=range(1,len(acc)+1)
plt.plot(epochs,acc,'bo',label='Trainingacc')
plt.plot(epochs,val_acc,'b',label='Validationacc')
plt.title('Training and validation accuracy with Data Augmentation')
plt.legend()
plt.figure()
plt.plot(epochs,loss,'bo',label='Training loss')
plt.plot(epochs,val_loss,'b',label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```