In [9]:
```python
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
import tensorflow as tf
import tensorflow_addons as tfa
from tensorflow.keras.utils import to_categorical
```

In [11]:
```python
(train_ds, train_labels), (test_ds, test_labels) =keras.datasets.cifar10.load_dat
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 752s 4us/step
```

In [15]:
```python
train_ds=train_ds[:500]
train_labels=train_labels[:500]
test_ds=test_ds[:500]
test_labels=test_labels[:500]
```

In [17]:
```python
train_ds[0].shape
```

Out[17]:
```
(32, 32, 3)
```

In [19]:
```python
train_labels = to_categorical(train_labels, num_classes=10)
test_labels = to_categorical(test_labels, num_classes=10)
```

In [21]:
```python
train_labels[0]
```

Out[21]:
```
array([0., 0., 0., 0., 0., 0., 1., 0., 0., 0.], dtype=float32)
```

In [25]:
```python
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg16 import preprocess_input
train_ds[0].shape
```

Out[25]:
```
(32, 32, 3)
```

In [27]:
```python
base_model = VGG16(weights="imagenet", include_top=False, input_shape=train_ds[0]
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-application
s/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 197s 3us/step
```

In [29]:
```python
base_model.trainable = False
## Preprocessing input
train_ds = preprocess_input(train_ds)
test_ds = preprocess_input(test_ds)
```

In [31]:
```python
## model details
base_model.summary()
```

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 32, 32, 3)]       0

 block1_conv1 (Conv2D)       (None, 32, 32, 64)        1792

 block1_conv2 (Conv2D)       (None, 32, 32, 64)        36928

 block1_pool (MaxPooling2D)  (None, 16, 16, 64)        0

 block2_conv1 (Conv2D)       (None, 16, 16, 128)       73856

 block2_conv2 (Conv2D)       (None, 16, 16, 128)       147584

 block2_pool (MaxPooling2D)  (None, 8, 8, 128)         0

 block3_conv1 (Conv2D)       (None, 8, 8, 256)         295168

 block3_conv2 (Conv2D)       (None, 8, 8, 256)         590080

 block3_conv3 (Conv2D)       (None, 8, 8, 256)         590080

 block3_pool (MaxPooling2D)  (None, 4, 4, 256)         0

 block4_conv1 (Conv2D)       (None, 4, 4, 512)         1180160

 block4_conv2 (Conv2D)       (None, 4, 4, 512)         2359808

 block4_conv3 (Conv2D)       (None, 4, 4, 512)         2359808

 block4_pool (MaxPooling2D)  (None, 2, 2, 512)         0

 block5_conv1 (Conv2D)       (None, 2, 2, 512)         2359808

 block5_conv2 (Conv2D)       (None, 2, 2, 512)         2359808

 block5_conv3 (Conv2D)       (None, 2, 2, 512)         2359808

 block5_pool (MaxPooling2D)  (None, 1, 1, 512)         0

=================================================================
Total params: 14714688 (56.13 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

In [33]:

```
#add our layers on top of this model
from tensorflow.keras import layers, models

flatten_layer = layers.Flatten()
dense_layer_1 = layers.Dense(50, activation='relu')
dense_layer_2 = layers.Dense (20, activation='relu')
prediction_layer = layers. Dense(10, activation='softmax')

model = models.Sequential([
    base_model,
    flatten_layer,
    dense_layer_1,
    dense_layer_2,
    prediction_layer
])
```

In [35]:
```
from tensorflow.keras.callbacks import EarlyStopping

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'],
)
```

In [39]:
```
es = EarlyStopping(monitor='val_accuracy', mode='max', patience=10, restore_best_
```

In [45]:
```
model.fit(train_ds, train_labels, epochs=5, validation_split=0.2, batch_size=32,
```

```
Epoch 1/5
13/13 [==============================] - 2s 137ms/step - loss: 0.9095 - accuracy:
0.7150 - val_loss: 3.8767 - val_accuracy: 0.2600
Epoch 2/5
13/13 [==============================] - 1s 102ms/step - loss: 0.7516 - accuracy:
0.7650 - val_loss: 3.9160 - val_accuracy: 0.2800
Epoch 3/5
13/13 [==============================] - 1s 104ms/step - loss: 0.6371 - accuracy:
0.8000 - val_loss: 3.9994 - val_accuracy: 0.2900
Epoch 4/5
13/13 [==============================] - 1s 104ms/step - loss: 0.5535 - accuracy:
0.8275 - val_loss: 4.0552 - val_accuracy: 0.2800
Epoch 5/5
13/13 [==============================] - 1s 105ms/step - loss: 0.4871 - accuracy:
0.8500 - val_loss: 4.1057 - val_accuracy: 0.2900
```

Out[45]:
```
<keras.src.callbacks.History at 0x1e778b54a10>
```

In [ ]: