In [1]:
```
import numpy as np
import cv2
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import keras
from   keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
import os
from   keras.models import Sequential
from   keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Averag
ePooling2D
```

In [2]:
```
test_data = 'test'
train_data = 'train'
```

In [3]:
```
train_datagen = ImageDataGenerator(rescale = 1./255)
test_datagen  = ImageDataGenerator(rescale = 1./255)
```

In [4]:
```
train_set = train_datagen.flow_from_directory(train_data,
                    target_size = (224,224),
                    batch_size = 32,
                    color_mode = 'rgb',
                    shuffle = True,
                    class_mode = 'categorical')
```

```
Found 14407 images belonging to 10 classes.
```

In [5]:
```
test_set = test_datagen.flow_from_directory(test_data,
                    target_size = (224,224),
                    batch_size = 32,
                    color_mode = 'rgb',
                    shuffle = True,
                    class_mode = 'categorical')
```

```
Found 3602 images belonging to 10 classes.
```

In [10]:
```
model = Sequential()


model.add(Conv2D(64, (3, 3), input_shape=(224,224,3), padding='same', a
ctivation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Flatten())
model.add(Dense(units = 4096, activation='relu'))
model.add(Dense(units = 4096, activation='relu'))
model.add(Dense(units = 10, activation='relu'))
```

In [11]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
conv2d_13 (Conv2D)           (None, 224, 224, 64)      1792      
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 224, 224, 64)      36928     
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 112, 112, 64)      0         
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 112, 112, 128)     73856     
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 112, 112, 128)     147584    
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 56, 56, 128)       0         
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 56, 56, 256)       295168    
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 56, 56, 256)       590080    
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 56, 56, 256)       590080    
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 28, 28, 256)       0         
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 28, 28, 512)       1180160   
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 28, 28, 512)       2359808   
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 28, 28, 512)       2359808   
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 14, 14, 512)       0         
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 14, 14, 512)       2359808   
_________________________________________________________________
conv2d_24 (Conv2D)           (None, 14, 14, 512)       2359808   
_________________________________________________________________
conv2d_25 (Conv2D)           (None, 14, 14, 512)       2359808   
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 7, 7, 512)         0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0         
_________________________________________________________________
```

```
dense_6 (Dense)              (None, 4096)              102764544
_________________________________________________________________
dense_7 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dense_8 (Dense)              (None, 10)                40970
=================================================================
Total params: 134,301,514
Trainable params: 134,301,514
Non-trainable params: 0

_________________________________________________________________
```

In [12]:
```
# Compile the model
model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics
=['accuracy'])
```

In [13]:
```
hist = model.fit(train_set, steps_per_epoch=200, epochs=10, validation_
data=test_set, validation_steps=100)
```

```
Epoch 1/10
  2/200 [..............................] - ETA: 1:12 - loss: 8.4314 - a
ccuracy: 0.1562WARNING:tensorflow:Callbacks method `on_train_batch_end`
is slow compared to the batch time (batch time: 0.1080s vs `on_train_ba
tch_end` time: 0.3140s). Check your callbacks.
200/200 [==============================] - 101s 507ms/step - loss: 8.97
76 - accuracy: 0.1118 - val_loss: 8.9330 - val_accuracy: 0.1078
Epoch 2/10
200/200 [==============================] - 101s 503ms/step - loss: 8.97
22 - accuracy: 0.1072 - val_loss: 8.8776 - val_accuracy: 0.1075
Epoch 3/10
200/200 [==============================] - 101s 503ms/step - loss: 8.71
84 - accuracy: 0.1156 - val_loss: 8.7266 - val_accuracy: 0.1119
Epoch 4/10
200/200 [==============================] - 101s 504ms/step - loss: 8.85
31 - accuracy: 0.1128 - val_loss: 8.8224 - val_accuracy: 0.1078
Epoch 5/10
169/200 [========================>.....] - ETA: 13s - loss: 8.8427 - ac
curacy: 0.1102

------------------------------------------------------------------------
```

```
----
KeyboardInterrupt                         Traceback (most recent call l
ast)
<ipython-input-13-a6dbe8732b70> in <module>
----> 1 hist = model.fit(train_set, steps_per_epoch=200, epochs=10, val
idation_data=test_set, validation_steps=100)

~\anaconda3\lib\site-packages\tensorflow\python\keras\engine\training.p
y in _method_wrapper(self, *args, **kwargs)
    106   def _method_wrapper(self, *args, **kwargs):
    107     if not self._in_multi_worker_mode():  # pylint: disable=pro
tected-access
--> 108       return method(self, *args, **kwargs)
    109
    110     # Running inside `run_distribute_coordinator` already.

~\anaconda3\lib\site-packages\tensorflow\python\keras\engine\training.p
y in fit(self, x, y, batch_size, epochs, verbose, callbacks, validation
_split, validation_data, shuffle, class_weight, sample_weight, initial_
epoch, steps_per_epoch, validation_steps, validation_batch_size, valida
tion_freq, max_queue_size, workers, use_multiprocessing)
   1101                 logs = tmp_logs  # No error, now safe to assign t
o logs.
   1102                 end_step = step + data_handler.step_increment
-> 1103                 callbacks.on_train_batch_end(end_step, logs)
   1104         epoch_logs = copy.copy(logs)
   1105

~\anaconda3\lib\site-packages\tensorflow\python\keras\callbacks.py in o
n_train_batch_end(self, batch, logs)
    438     """
    439     if self._should_call_train_batch_hooks:
--> 440       self._call_batch_hook(ModeKeys.TRAIN, 'end', batch, logs=
logs)
    441
    442   def on_test_batch_begin(self, batch, logs=None):

~\anaconda3\lib\site-packages\tensorflow\python\keras\callbacks.py in _
call_batch_hook(self, mode, hook, batch, logs)
```

```
    287       self._call_batch_begin_hook(mode, batch, logs)
    288     elif hook == 'end':
--> 289       self._call_batch_end_hook(mode, batch, logs)
    290     else:
    291       raise ValueError('Unrecognized hook: {}'.format(hook))

~\anaconda3\lib\site-packages\tensorflow\python\keras\callbacks.py in _
call_batch_end_hook(self, mode, batch, logs)
    307       batch_time = time.time() - self._batch_start_time
    308 
--> 309     self._call_batch_hook_helper(hook_name, batch, logs)
    310 
    311     if self._check_timing:

~\anaconda3\lib\site-packages\tensorflow\python\keras\callbacks.py in _
call_batch_hook_helper(self, hook_name, batch, logs)
    340       hook = getattr(callback, hook_name)
    341       if getattr(callback, '_supports_tf_logs', False):
--> 342         hook(batch, logs)
    343       else:
    344         if numpy_logs is None:  # Only convert once.

~\anaconda3\lib\site-packages\tensorflow\python\keras\callbacks.py in o
n_train_batch_end(self, batch, logs)
    959 
    960   def on_train_batch_end(self, batch, logs=None):
--> 961     self._batch_update_progbar(batch, logs)
    962 
    963   def on_test_batch_end(self, batch, logs=None):

~\anaconda3\lib\site-packages\tensorflow\python\keras\callbacks.py in _
batch_update_progbar(self, batch, logs)
   1014     if self.verbose == 1:
   1015       # Only block async when verbose = 1.
-> 1016       logs = tf_utils.to_numpy_or_python_type(logs)
   1017       self.progbar.update(self.seen, list(logs.items()), finali
ze=False)
   1018 
```

```
~\anaconda3\lib\site-packages\tensorflow\python\keras\utils\tf_utils.py
 in to_numpy_or_python_type(tensors)
    535       return t  # Don't turn ragged or sparse tensors to NumPy.
    536 
--> 537   return nest.map_structure(_to_single_numpy_or_python_type, te
nsors)
    538 
    539 

~\anaconda3\lib\site-packages\tensorflow\python\util\nest.py in map_str
ucture(func, *structure, **kwargs)
    633 
    634   return pack_sequence_as(
--> 635       structure[0], [func(*x) for x in entries],
    636       expand_composites=expand_composites)
    637 

~\anaconda3\lib\site-packages\tensorflow\python\util\nest.py in <listco
mp>(.0)
    633 
    634   return pack_sequence_as(
--> 635       structure[0], [func(*x) for x in entries],
    636       expand_composites=expand_composites)
    637 

~\anaconda3\lib\site-packages\tensorflow\python\keras\utils\tf_utils.py
 in _to_single_numpy_or_python_type(t)
    531   def _to_single_numpy_or_python_type(t):
    532     if isinstance(t, ops.Tensor):
--> 533       x = t.numpy()
    534       return x.item() if np.ndim(x) == 0 else x
    535     return t  # Don't turn ragged or sparse tensors to NumPy.

~\anaconda3\lib\site-packages\tensorflow\python\framework\ops.py in num
py(self)
   1061     """
   1062     # TODO(slebedev): Consider avoiding a copy for non-CPU or r
emote tensors.
-> 1063     maybe_arr = self._numpy()  # pylint: disable=protected-acce
```

```
ss
   1064     return maybe_arr.copy() if isinstance(maybe_arr, np.ndarray
) else maybe_arr
   1065

~\anaconda3\lib\site-packages\tensorflow\python\framework\ops.py in _nu
mpy(self)
   1027   def _numpy(self):
   1028     try:
-> 1029       return self._numpy_internal()
   1030     except core._NotOkStatusException as e:  # pylint: disable=
protected-access
   1031       six.raise_from(core._status_to_exception(e.code, e.messag
e), None)  # pylint: disable=protected-access

KeyboardInterrupt:
```

```
In [ ]:
```