Garbage Classification

This work is the capstone project of the course 'Advanced Data Science' on Coursera. Our task is to select a dataset and use it to show the understanding of the workflow a Machine Learning project. In this notebook we:

- Load, explore and preprocess the data
- Define, create and train a simple Logistic Regression Model
- Define, create and train a deep Convolutional Model
- Train the Convolutional Model with normalized data
- Define, calculate and explain different metrics to evaluate the performance of each model

The dataset is called 'Garbage Classification' (https://www.kaggle.com/asdasdasasdas/garbage-classification/ (http://)) and contains images of different kind of garbage, our goal is to correctly identify each image.

In [2]:
```
import os
import glob
import matplotlib.pyplot as plt
import numpy as np
```

Data Loading

Let's load the dataset and start exploring its features.

In [3]:
```
BASE_DIR = 'datasets'
IMG_DIR = f'{BASE_DIR}/Garbage classification'

path_list = glob.glob(os.path.join(IMG_DIR, '*/*.jpg'))
num_samples = len(path_list)
print(f'Found {num_samples} images')
```

```
Found 2527 images
```

The dataset contains of 2527 images from 6 different classes: *cardboard*, *glass*, *metal*, *paper*, *plastic* and *trash*. The images are 3 channels RGB with shape 512×384, but for our purposes they will be cropped to be square with dimension 384×384.

Data Augmentation

Since the dataset is relatively small we decided not to split into train, validation and test images beforehand. In fact, this will reduce the dataset size even more which could easily lead to overfitting. Instead we keep 10% as validation data and the rest is used for training, and to increase the performance on new samples, we will augment the training dataset by: flipping, zooming, shearing and shifting the images. This also should help prevent overfitting and give us the opportunity to build a deeper model.

In [4]:
```
from keras.preprocessing.image import ImageDataGenerator

img_size = (384, 384)
batch_size = 32

tr=ImageDataGenerator(horizontal_flip=True,
                         vertical_flip=True,
                         validation_split=0.1,
                         shear_range = 0.1,
                         zoom_range = 0.1,
                         width_shift_range = 0.1,
                         height_shift_range = 0.1)

val=ImageDataGenerator(validation_split = 0.1)

train_generator=tr.flow_from_directory(IMG_DIR,
                                       target_size=img_size,
                                       batch_size=batch_size,
                                       class_mode='categorical',
                                       subset='training')

val_generator=val.flow_from_directory(IMG_DIR,
                                      target_size=img_size,
                                      batch_size=batch_size,
                                      class_mode='categorical',
                                      subset='validation')
```

```
Found 2276 images belonging to 6 classes.
Found 251 images belonging to 6 classes.
```

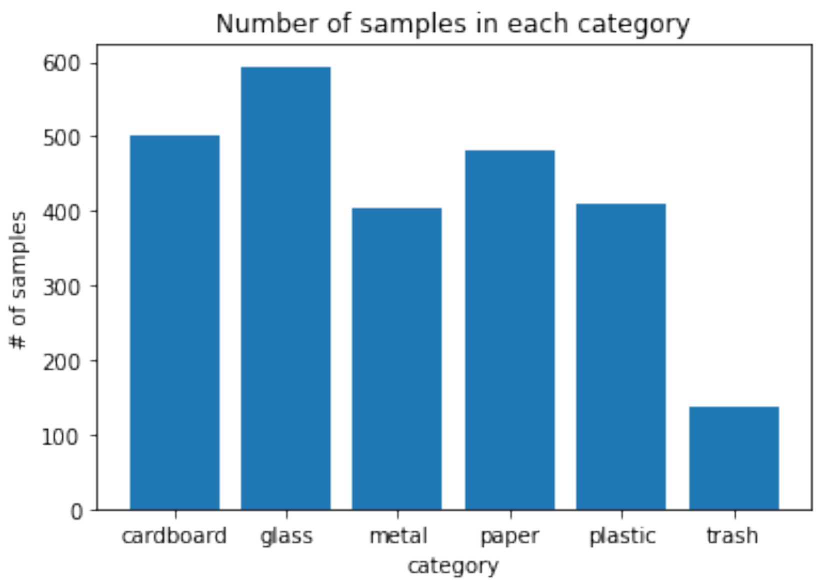
Balance Dataset

This dataset is not balanced, meaning that there are different numbers of samples for each category. This could lead to inaccuracies during the training phase as the model will be biased towards predicting the classes which occur more.

In [5]:
```python
import pandas as pd

samples = pd.read_csv(f'{BASE_DIR}/zero-indexed-files.txt', sep = '
', names = ['path', 'label'])
unique, counts = np.unique(samples['label'], return_counts=True)

class_to_id = train_generator.class_indices
id_to_class = {k: v for v, k in class_to_id.items()}
print('Number of samples: ')
for count, category in zip(counts, class_to_id.keys()):
    print(f'{count} in {category}')
```

```
Number of samples:
501 in cardboard
594 in glass
403 in metal
482 in paper
410 in plastic
137 in trash
```

In [6]:
```python
plt.bar(list(class_to_id.keys()), counts)
plt.title('Number of samples in each category')
plt.xlabel('category')
plt.ylabel('# of samples')
plt.show()
```



As we can clearly see there is a relevant imbalance between the number of samples in the category "trash" and the others. To present less biased data to our model, we calculate `class_weigths` which we will pass to our `fit` function during the training phase. This will ensure that the frequency of occurence for samples of each class is equal to the others.

```
In [7]: weights = [(1 / category_count) * (num_samples) / len(unique) for c
        ategory_count in counts]
        class_weights = {k: v for k, v in zip(unique, weights)}
        class_weights
```

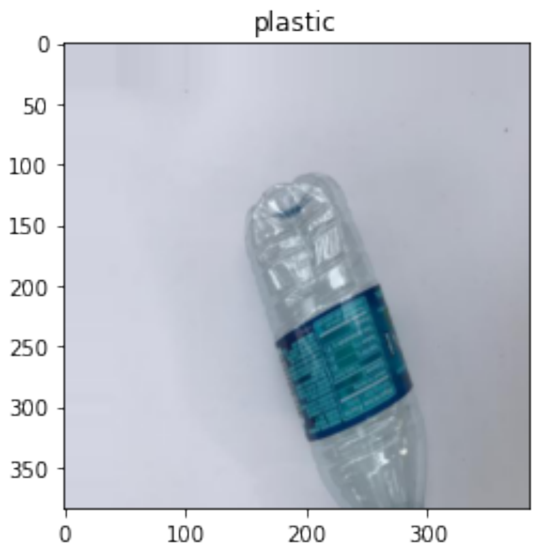
```
Out[7]: {0: 0.8406520292747838,
         1: 0.7090347923681257,
         2: 1.0450785773366418,
         3: 0.873789764868603,
         4: 1.0272357723577235,
         5: 3.0742092457420926}
```

These weights are inversely proportional to the size of each class and will smooth out the difference in occurence frequency.

Show some samples

This is one of the samples which we will work with.

```
In [8]: x, y = train_generator.next()
        category = np.argmax(y[0])

        plt.title(id_to_class[category])
        plt.imshow((x[0]).astype('int'))
        plt.show()
```



Models

Logistic Regression model

We now build the first model. It is a simple Logistic Regression Model with one hidden layer with 6 neurons. The chosen activation function is ***softmax*** in order to return a confidence score for the prediction.

In [9]:
```
from keras import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.callbacks import ModelCheckpoint


def logistic_regression_model():

    model = Sequential()
    model.add(Flatten())
    model.add(Dense(6,activation='softmax'))

    return model
```

Define training function

In [10]:
```
from keras.callbacks import ModelCheckpoint
from keras.callbacks import ReduceLROnPlateau

def train(model, train_gen, val_gen, epochs, best_model_name='best_
model.h5'):

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

    checkpoint = ModelCheckpoint(best_model_name,
                                 monitor='val_acc',
                                 verbose=1,
                                 save_best_only=True,
                                 mode='max')

    reduce_lr = ReduceLROnPlateau(monitor='val_acc',
                                  mode='max')

    history = model.fit(train_gen,
                        epochs=epochs,
                        validation_data=val_gen,
                        class_weight=class_weights,
                        workers = 4,
                        callbacks=[checkpoint, reduce_lr])
    return history
```

Compile and train the model

Since the model is shallow, 20 epochs are enough to train the model and prevent overfitting

In [11]:
```
lr = logistic_regression_model()
lr_history = train(lr, train_generator, val_generator, epochs = 20,
best_model_name = 'best_simple_model.h5')
```

```
Epoch 1/20
72/72 [==============================] - ETA: 0s - loss: 34663.367
2 - acc: 0.2140
Epoch 00001: val_acc improved from -inf to 0.26693, saving model t
o best_simple_model.h5
72/72 [==============================] - 81s 1s/step - loss: 34663
.3672 - acc: 0.2140 - val_loss: 7593.3906 - val_acc: 0.2669
Epoch 2/20
72/72 [==============================] - ETA: 0s - loss: 17100.267
6 - acc: 0.2438
Epoch 00002: val_acc improved from 0.26693 to 0.29084, saving mode
l to best_simple_model.h5
72/72 [==============================] - 82s 1s/step - loss: 17100
.2676 - acc: 0.2438 - val_loss: 11652.9092 - val_acc: 0.2908
Epoch 3/20
72/72 [==============================] - ETA: 0s - loss: 8300.9717
- acc: 0.2882
Epoch 00003: val_acc improved from 0.29084 to 0.35060, saving mode
l to best_simple_model.h5
72/72 [==============================] - 81s 1s/step - loss: 8300.
9717 - acc: 0.2882 - val_loss: 4580.7803 - val_acc: 0.3506
Epoch 4/20
72/72 [==============================] - ETA: 0s - loss: 11238.280
3 - acc: 0.2790
Epoch 00004: val_acc did not improve from 0.35060
72/72 [==============================] - 81s 1s/step - loss: 11238
.2803 - acc: 0.2790 - val_loss: 11065.3389 - val_acc: 0.2789
Epoch 5/20
72/72 [==============================] - ETA: 0s - loss: 15006.248
0 - acc: 0.2658
Epoch 00005: val_acc did not improve from 0.35060
72/72 [==============================] - 81s 1s/step - loss: 15006
.2480 - acc: 0.2658 - val_loss: 19536.3945 - val_acc: 0.2869
Epoch 6/20
72/72 [==============================] - ETA: 0s - loss: 16600.011
7 - acc: 0.2632
Epoch 00006: val_acc did not improve from 0.35060
72/72 [==============================] - 84s 1s/step - loss: 16600
.0117 - acc: 0.2632 - val_loss: 8111.8447 - val_acc: 0.3466
Epoch 7/20
72/72 [==============================] - ETA: 0s - loss: 16070.081
1 - acc: 0.2592
Epoch 00007: val_acc did not improve from 0.35060
72/72 [==============================] - 80s 1s/step - loss: 16070
.0811 - acc: 0.2592 - val_loss: 27188.5078 - val_acc: 0.2430
Epoch 8/20
```

```
72/72 [==============================] - ETA: 0s - loss: 13541.084
0 - acc: 0.2926
Epoch 00008: val_acc did not improve from 0.35060
72/72 [==============================] - 77s 1s/step - loss: 13541
.0840 - acc: 0.2926 - val_loss: 10172.9053 - val_acc: 0.2629
Epoch 9/20
72/72 [==============================] - ETA: 0s - loss: 10730.418
0 - acc: 0.2944
Epoch 00009: val_acc did not improve from 0.35060
72/72 [==============================] - 78s 1s/step - loss: 10730
.4180 - acc: 0.2944 - val_loss: 8423.9766 - val_acc: 0.3466
Epoch 10/20
72/72 [==============================] - ETA: 0s - loss: 9264.6514
- acc: 0.3071
Epoch 00010: val_acc did not improve from 0.35060
72/72 [==============================] - 78s 1s/step - loss: 9264.
6514 - acc: 0.3071 - val_loss: 14027.7100 - val_acc: 0.2430
Epoch 11/20
72/72 [==============================] - ETA: 0s - loss: 12140.317
4 - acc: 0.2856
Epoch 00011: val_acc did not improve from 0.35060
72/72 [==============================] - 77s 1s/step - loss: 12140
.3174 - acc: 0.2856 - val_loss: 19623.3027 - val_acc: 0.0996
Epoch 12/20
72/72 [==============================] - ETA: 0s - loss: 12037.948
2 - acc: 0.2764
Epoch 00012: val_acc did not improve from 0.35060
72/72 [==============================] - 75s 1s/step - loss: 12037
.9482 - acc: 0.2764 - val_loss: 14415.4092 - val_acc: 0.2550
Epoch 13/20
71/72 [============================>.] - ETA: 0s - loss: 13802.185
5 - acc: 0.2672
Epoch 00013: val_acc did not improve from 0.35060
72/72 [==============================] - 74s 1s/step - loss: 13794
.1211 - acc: 0.2676 - val_loss: 13597.6553 - val_acc: 0.2590
Epoch 14/20
72/72 [==============================] - ETA: 0s - loss: 4468.7344
- acc: 0.4042
Epoch 00014: val_acc did not improve from 0.35060
72/72 [==============================] - 76s 1s/step - loss: 4468.
7344 - acc: 0.4042 - val_loss: 4313.2324 - val_acc: 0.3108
Epoch 15/20
72/72 [==============================] - ETA: 0s - loss: 3343.1567
- acc: 0.4073
Epoch 00015: val_acc improved from 0.35060 to 0.39442, saving mode
l to best_simple_model.h5
72/72 [==============================] - 75s 1s/step - loss: 3343.
1567 - acc: 0.4073 - val_loss: 3073.5664 - val_acc: 0.3944
Epoch 16/20
72/72 [==============================] - ETA: 0s - loss: 3074.9463
- acc: 0.4139
Epoch 00016: val_acc did not improve from 0.39442
72/72 [==============================] - 75s 1s/step - loss: 3074.
9463 - acc: 0.4139 - val_loss: 3094.8142 - val_acc: 0.3825
Epoch 17/20
72/72 [==============================] - ETA: 0s - loss: 2808.0471
- acc: 0.3959
```

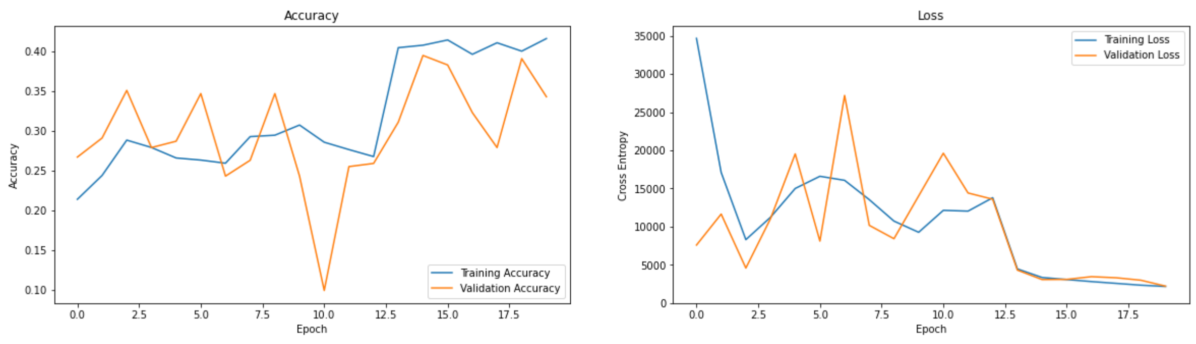
```
Epoch 00017: val_acc did not improve from 0.39442
72/72 [==============================] - 72s 1000ms/step - loss: 2
808.0471 - acc: 0.3959 - val_loss: 3449.6443 - val_acc: 0.3227
Epoch 18/20
72/72 [==============================] - ETA: 0s - loss: 2572.7454
- acc: 0.4104
Epoch 00018: val_acc did not improve from 0.39442
72/72 [==============================] - 73s 1s/step - loss: 2572.
7454 - acc: 0.4104 - val_loss: 3307.7419 - val_acc: 0.2789
Epoch 19/20
72/72 [==============================] - ETA: 0s - loss: 2336.6018
- acc: 0.3998
Epoch 00019: val_acc did not improve from 0.39442
72/72 [==============================] - 75s 1s/step - loss: 2336.
6018 - acc: 0.3998 - val_loss: 2986.0984 - val_acc: 0.3904
Epoch 20/20
72/72 [==============================] - ETA: 0s - loss: 2168.9084
- acc: 0.4156
Epoch 00020: val_acc did not improve from 0.39442
72/72 [==============================] - 74s 1s/step - loss: 2168.
9084 - acc: 0.4156 - val_loss: 2211.4790 - val_acc: 0.3426
```

Results analysis: Logistic Regression

The first step to get an idea about the capabilities of the model is to see the accuracy and the loss during the training and validation phase. This way, we can detect overfitting at a glance and determine if the lerning rate and the number of epochs were adequate.

In [11]:
```
def plot_history(history):

    acc = history.history['acc']
    val_acc = history.history['val_acc']

    loss = history.history['loss']
    val_loss = history.history['val_loss']

    # Plot accuracy
    plt.figure(figsize=(20, 5))
    plt.subplot(1, 2, 1)
    plt.plot(acc, label='Training Accuracy')
    plt.plot(val_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.title('Accuracy')

    # Plot loss
    plt.subplot(1, 2, 2)
    plt.plot(loss, label='Training Loss')
    plt.plot(val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.xlabel('Epoch')
    plt.ylabel('Cross Entropy')
    plt.ylim([0,max(plt.ylim())])
    plt.title('Loss')
    plt.show()
```

In [13]:
```
plot_history(lr_history)
```



We can then evaluate the model using different metrics, in this case we used the **confusion matrix** and the **F1 score**.

Since the dataset is quite small, we will use it as a whole to make the predictions. This could lead to inaccurate results in case of overfitting, but since we augmented the dataset in the training phase, we prevent this situation to occur.

```
In [12]: test_generator = ImageDataGenerator().flow_from_directory(IMG_DIR,
                                                          target_size=img_size,
                                                          batch_size=1,
                                                          class_mode='categorical')
```

```
Found 2527 images belonging to 6 classes.
```

Let's define a function which returns the true labels and the predictions and store them in `ys` and `preds` .

```
In [13]: def evaluate(m, test_gen):
             preds = []
             ys = []
             for i in range(len(test_generator)):
                 pred = np.argmax(m.predict(test_gen[i][0]))
                 preds.append(pred)
                 y = np.argmax(test_gen[i][1])
                 ys.append(y)
             return ys, preds
```

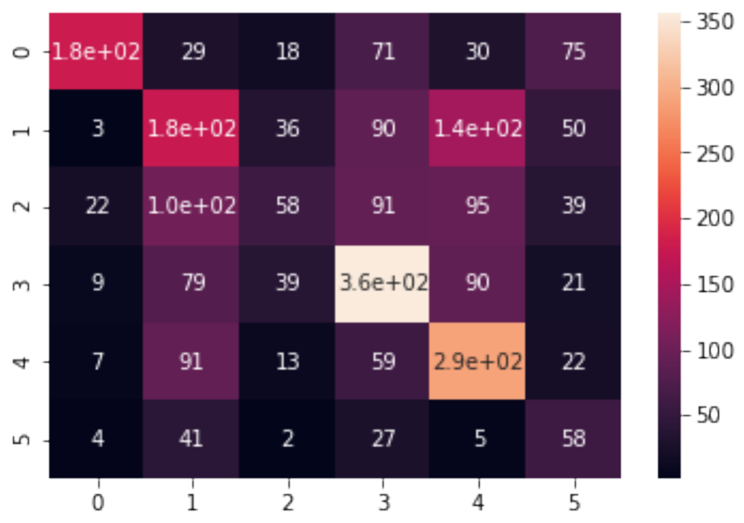
```
In [16]: lr_ys, lr_preds = evaluate(lr, test_genenerator)
```

Confusion matrix

The **confusion matrix** contains the predicted classes as rows, and the target classes as columns. This way, we can see which classes the model predicts best and which one it struggles with. We want to maximise the elements on the diagonal as they are the correct predictions.

```
In [15]: from sklearn.metrics import confusion_matrix
         import seaborn as sn

         def confusion(ys, preds):
             matrix = confusion_matrix(ys, preds)
             sn.heatmap(matrix, annot=True)
```

```
In [18]: confusion(lr_ys, lr_preds)
```



F1 Score

The **F1 score** is defined as

$$2 * \frac{precision * recall}{precision + recall}$$

where

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad recall = \frac{True\ Positives}{True\ Positives + False\ Negative}$$

This score balances the measure of the exactness ($precision$) and completeness ($recall$) of the model and the closer to 1 it is, the better the model performance.

```
In [16]: from sklearn.metrics import f1_score

         def f1(ys, preds):
             score = f1_score(ys, preds, average='weighted')
             print(f'F1 score is: {score}')
             return score
```

```
In [20]: f1_lr = f1(lr_ys, lr_preds)
```

```
F1 score is: 0.4355848113706974
```

Convolutional Model

Let's build the second model. It has 4 convolutional layers and MaxPooling layers which constitute convolutional section of the model, where the features are extracted from the images. Then 3 fully connected layers with Dropout make up the classificsation section, these layers will classify the images based on the feature previously extracted.

In [17]:
```
from keras.layers import Dense
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout


def CNN_model():

    model = Sequential()

    #Convolutional layers
    model.add(Conv2D(16,(3,3), padding='same',activation='relu', in
put_shape=(*img_size,3)))
    model.add(MaxPooling2D(pool_size=2))

    model.add(Conv2D(32,(3,3), padding='same',activation='relu'))
    model.add(MaxPooling2D(pool_size=2))

    model.add(Conv2D(64,(3,3), padding='same',activation='relu'))
    model.add(MaxPooling2D(pool_size=2))

    model.add(Conv2D(32,(3,3), padding='same',activation='relu'))
    model.add(MaxPooling2D(pool_size=2))

    #Classification layers
    model.add(Flatten())

    model.add(Dense(64,activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(32,activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(6,activation='softmax'))

    return model
```

Train convolutional model

In [23]:
```
cnn = CNN_model()
cnn_history = train(cnn, train_generator, val_generator, epochs = 8
0, best_model_name='best_cnn_model.h5')
```

```
Epoch 1/80
72/72 [==============================] - ETA: 0s - loss: 5.9379 -
acc: 0.2017
Epoch 00001: val_acc improved from -inf to 0.24701, saving model t
o best_cnn_model.h5
72/72 [==============================] - 76s 1s/step - loss: 5.937
9 - acc: 0.2017 - val_loss: 1.7731 - val_acc: 0.2470
Epoch 2/80
72/72 [==============================] - ETA: 0s - loss: 1.7993 -
acc: 0.2399
Epoch 00002: val_acc improved from 0.24701 to 0.31076, saving mode
```

```
l to best_cnn_model.h5
72/72 [==============================] - 80s 1s/step - loss: 1.799
3 - acc: 0.2399 - val_loss: 1.7631 - val_acc: 0.3108
Epoch 3/80
72/72 [==============================] - ETA: 0s - loss: 1.7598 -
acc: 0.2922
Epoch 00003: val_acc did not improve from 0.31076
72/72 [==============================] - 82s 1s/step - loss: 1.759
8 - acc: 0.2922 - val_loss: 1.7848 - val_acc: 0.2311
Epoch 4/80
72/72 [==============================] - ETA: 0s - loss: 1.7896 -
acc: 0.2192
Epoch 00004: val_acc did not improve from 0.31076
72/72 [==============================] - 82s 1s/step - loss: 1.789
6 - acc: 0.2192 - val_loss: 1.7710 - val_acc: 0.2231
Epoch 5/80
72/72 [==============================] - ETA: 0s - loss: 1.7605 -
acc: 0.2399
Epoch 00005: val_acc did not improve from 0.31076
72/72 [==============================] - 81s 1s/step - loss: 1.760
5 - acc: 0.2399 - val_loss: 1.7240 - val_acc: 0.2789
Epoch 6/80
72/72 [==============================] - ETA: 0s - loss: 1.7762 -
acc: 0.2540
Epoch 00006: val_acc did not improve from 0.31076
72/72 [==============================] - 84s 1s/step - loss: 1.776
2 - acc: 0.2540 - val_loss: 1.7173 - val_acc: 0.3108
Epoch 7/80
72/72 [==============================] - ETA: 0s - loss: 1.7458 -
acc: 0.2913
Epoch 00007: val_acc did not improve from 0.31076
72/72 [==============================] - 84s 1s/step - loss: 1.745
8 - acc: 0.2913 - val_loss: 1.6824 - val_acc: 0.2869
Epoch 8/80
72/72 [==============================] - ETA: 0s - loss: 1.7180 -
acc: 0.3076
Epoch 00008: val_acc did not improve from 0.31076
72/72 [==============================] - 85s 1s/step - loss: 1.718
0 - acc: 0.3076 - val_loss: 1.6821 - val_acc: 0.2908
Epoch 9/80
72/72 [==============================] - ETA: 0s - loss: 1.6986 -
acc: 0.3067
Epoch 00009: val_acc improved from 0.31076 to 0.40239, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 85s 1s/step - loss: 1.698
6 - acc: 0.3067 - val_loss: 1.6016 - val_acc: 0.4024
Epoch 10/80
72/72 [==============================] - ETA: 0s - loss: 1.6349 -
acc: 0.3484
Epoch 00010: val_acc did not improve from 0.40239
72/72 [==============================] - 86s 1s/step - loss: 1.634
9 - acc: 0.3484 - val_loss: 1.5305 - val_acc: 0.3984
Epoch 11/80
72/72 [==============================] - ETA: 0s - loss: 1.6033 -
acc: 0.3612
Epoch 00011: val_acc did not improve from 0.40239
72/72 [==============================] - 84s 1s/step - loss: 1.603
```

```
3 - acc: 0.3612 - val_loss: 1.6616 - val_acc: 0.3347
Epoch 12/80
72/72 [==============================] - ETA: 0s - loss: 1.5912 -
acc: 0.3748
Epoch 00012: val_acc did not improve from 0.40239
72/72 [==============================] - 88s 1s/step - loss: 1.591
2 - acc: 0.3748 - val_loss: 1.5990 - val_acc: 0.3307
Epoch 13/80
72/72 [==============================] - ETA: 0s - loss: 1.6132 -
acc: 0.3440
Epoch 00013: val_acc did not improve from 0.40239
72/72 [==============================] - 87s 1s/step - loss: 1.613
2 - acc: 0.3440 - val_loss: 1.5749 - val_acc: 0.3028
Epoch 14/80
72/72 [==============================] - ETA: 0s - loss: 1.5517 -
acc: 0.3862
Epoch 00014: val_acc improved from 0.40239 to 0.41833, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 86s 1s/step - loss: 1.551
7 - acc: 0.3862 - val_loss: 1.4744 - val_acc: 0.4183
Epoch 15/80
72/72 [==============================] - ETA: 0s - loss: 1.5038 -
acc: 0.3959
Epoch 00015: val_acc did not improve from 0.41833
72/72 [==============================] - 87s 1s/step - loss: 1.503
8 - acc: 0.3959 - val_loss: 1.4851 - val_acc: 0.4024
Epoch 16/80
72/72 [==============================] - ETA: 0s - loss: 1.4789 -
acc: 0.4134
Epoch 00016: val_acc did not improve from 0.41833
72/72 [==============================] - 82s 1s/step - loss: 1.478
9 - acc: 0.4134 - val_loss: 1.5583 - val_acc: 0.3466
Epoch 17/80
72/72 [==============================] - ETA: 0s - loss: 1.4663 -
acc: 0.4134
Epoch 00017: val_acc did not improve from 0.41833
72/72 [==============================] - 80s 1s/step - loss: 1.466
3 - acc: 0.4134 - val_loss: 1.4520 - val_acc: 0.3785
Epoch 18/80
72/72 [==============================] - ETA: 0s - loss: 1.4402 -
acc: 0.4192
Epoch 00018: val_acc did not improve from 0.41833
72/72 [==============================] - 78s 1s/step - loss: 1.440
2 - acc: 0.4192 - val_loss: 1.4772 - val_acc: 0.3944
Epoch 19/80
72/72 [==============================] - ETA: 0s - loss: 1.3948 -
acc: 0.4240
Epoch 00019: val_acc did not improve from 0.41833
72/72 [==============================] - 77s 1s/step - loss: 1.394
8 - acc: 0.4240 - val_loss: 1.4936 - val_acc: 0.3347
Epoch 20/80
72/72 [==============================] - ETA: 0s - loss: 1.3664 -
acc: 0.4464
Epoch 00020: val_acc did not improve from 0.41833
72/72 [==============================] - 76s 1s/step - loss: 1.366
4 - acc: 0.4464 - val_loss: 1.4176 - val_acc: 0.3546
Epoch 21/80
```

```
72/72 [==============================] - ETA: 0s - loss: 1.3777 -
acc: 0.4367
Epoch 00021: val_acc did not improve from 0.41833
72/72 [==============================] - 75s 1s/step - loss: 1.377
7 - acc: 0.4367 - val_loss: 1.4222 - val_acc: 0.3745
Epoch 22/80
72/72 [==============================] - ETA: 0s - loss: 1.3517 -
acc: 0.4464
Epoch 00022: val_acc did not improve from 0.41833
72/72 [==============================] - 77s 1s/step - loss: 1.351
7 - acc: 0.4464 - val_loss: 1.3962 - val_acc: 0.3984
Epoch 23/80
72/72 [==============================] - ETA: 0s - loss: 1.3139 -
acc: 0.4829
Epoch 00023: val_acc improved from 0.41833 to 0.46215, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 75s 1s/step - loss: 1.313
9 - acc: 0.4829 - val_loss: 1.3328 - val_acc: 0.4622
Epoch 24/80
72/72 [==============================] - ETA: 0s - loss: 1.2873 -
acc: 0.4780
Epoch 00024: val_acc did not improve from 0.46215
72/72 [==============================] - 75s 1s/step - loss: 1.287
3 - acc: 0.4780 - val_loss: 1.3996 - val_acc: 0.4143
Epoch 25/80
72/72 [==============================] - ETA: 0s - loss: 1.3309 -
acc: 0.4675
Epoch 00025: val_acc improved from 0.46215 to 0.48207, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 75s 1s/step - loss: 1.330
9 - acc: 0.4675 - val_loss: 1.2880 - val_acc: 0.4821
Epoch 26/80
72/72 [==============================] - ETA: 0s - loss: 1.2947 -
acc: 0.4798
Epoch 00026: val_acc did not improve from 0.48207
72/72 [==============================] - 74s 1s/step - loss: 1.294
7 - acc: 0.4798 - val_loss: 1.3196 - val_acc: 0.4701
Epoch 27/80
72/72 [==============================] - ETA: 0s - loss: 1.3175 -
acc: 0.4750
Epoch 00027: val_acc did not improve from 0.48207
72/72 [==============================] - 73s 1s/step - loss: 1.317
5 - acc: 0.4750 - val_loss: 1.7464 - val_acc: 0.3745
Epoch 28/80
72/72 [==============================] - ETA: 0s - loss: 1.3976 -
acc: 0.4473
Epoch 00028: val_acc did not improve from 0.48207
72/72 [==============================] - 75s 1s/step - loss: 1.397
6 - acc: 0.4473 - val_loss: 1.2759 - val_acc: 0.4701
Epoch 29/80
72/72 [==============================] - ETA: 0s - loss: 1.2521 -
acc: 0.5009
Epoch 00029: val_acc did not improve from 0.48207
72/72 [==============================] - 72s 1s/step - loss: 1.252
1 - acc: 0.5009 - val_loss: 1.2341 - val_acc: 0.4821
Epoch 30/80
72/72 [==============================] - ETA: 0s - loss: 1.2276 -
```

```
acc: 0.4947
Epoch 00030: val_acc did not improve from 0.48207
72/72 [==============================] - 73s 1s/step - loss: 1.227
6 - acc: 0.4947 - val_loss: 1.2424 - val_acc: 0.4821
Epoch 31/80
72/72 [==============================] - ETA: 0s - loss: 1.2676 -
acc: 0.5048
Epoch 00031: val_acc did not improve from 0.48207
72/72 [==============================] - 73s 1s/step - loss: 1.267
6 - acc: 0.5048 - val_loss: 1.3349 - val_acc: 0.4462
Epoch 32/80
72/72 [==============================] - ETA: 0s - loss: 1.2315 -
acc: 0.5062
Epoch 00032: val_acc did not improve from 0.48207
72/72 [==============================] - 73s 1s/step - loss: 1.231
5 - acc: 0.5062 - val_loss: 1.2507 - val_acc: 0.4622
Epoch 33/80
72/72 [==============================] - ETA: 0s - loss: 1.2003 -
acc: 0.5158
Epoch 00033: val_acc improved from 0.48207 to 0.49402, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 73s 1s/step - loss: 1.200
3 - acc: 0.5158 - val_loss: 1.2528 - val_acc: 0.4940
Epoch 34/80
72/72 [==============================] - ETA: 0s - loss: 1.2471 -
acc: 0.5136
Epoch 00034: val_acc did not improve from 0.49402
72/72 [==============================] - 74s 1s/step - loss: 1.247
1 - acc: 0.5136 - val_loss: 1.3693 - val_acc: 0.4502
Epoch 35/80
72/72 [==============================] - ETA: 0s - loss: 1.1685 -
acc: 0.5272
Epoch 00035: val_acc improved from 0.49402 to 0.49801, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 72s 1s/step - loss: 1.168
5 - acc: 0.5272 - val_loss: 1.2615 - val_acc: 0.4980
Epoch 36/80
72/72 [==============================] - ETA: 0s - loss: 1.2020 -
acc: 0.5224
Epoch 00036: val_acc did not improve from 0.49801
72/72 [==============================] - 71s 991ms/step - loss: 1.
2020 - acc: 0.5224 - val_loss: 1.2016 - val_acc: 0.4980
Epoch 37/80
72/72 [==============================] - ETA: 0s - loss: 1.1332 -
acc: 0.5409
Epoch 00037: val_acc did not improve from 0.49801
72/72 [==============================] - 72s 994ms/step - loss: 1.
1332 - acc: 0.5409 - val_loss: 1.2218 - val_acc: 0.4781
Epoch 38/80
72/72 [==============================] - ETA: 0s - loss: 1.1591 -
acc: 0.5246
Epoch 00038: val_acc did not improve from 0.49801
72/72 [==============================] - 72s 1s/step - loss: 1.159
1 - acc: 0.5246 - val_loss: 1.3728 - val_acc: 0.4223
Epoch 39/80
72/72 [==============================] - ETA: 0s - loss: 1.2105 -
acc: 0.5325
```

```
Epoch 00039: val_acc improved from 0.49801 to 0.51394, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 71s 993ms/step - loss: 1.
2105 - acc: 0.5325 - val_loss: 1.2104 - val_acc: 0.5139
Epoch 40/80
72/72 [==============================] - ETA: 0s - loss: 1.2354 -
acc: 0.5088
Epoch 00040: val_acc improved from 0.51394 to 0.52590, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 73s 1s/step - loss: 1.235
4 - acc: 0.5088 - val_loss: 1.1557 - val_acc: 0.5259
Epoch 41/80
72/72 [==============================] - ETA: 0s - loss: 1.1496 -
acc: 0.5584
Epoch 00041: val_acc did not improve from 0.52590
72/72 [==============================] - 72s 1s/step - loss: 1.149
6 - acc: 0.5584 - val_loss: 1.1476 - val_acc: 0.5179
Epoch 42/80
72/72 [==============================] - ETA: 0s - loss: 1.1321 -
acc: 0.5514
Epoch 00042: val_acc did not improve from 0.52590
72/72 [==============================] - 72s 997ms/step - loss: 1.
1321 - acc: 0.5514 - val_loss: 1.1657 - val_acc: 0.5139
Epoch 43/80
72/72 [==============================] - ETA: 0s - loss: 1.1218 -
acc: 0.5598
Epoch 00043: val_acc did not improve from 0.52590
72/72 [==============================] - 71s 987ms/step - loss: 1.
1218 - acc: 0.5598 - val_loss: 1.4478 - val_acc: 0.4462
Epoch 44/80
72/72 [==============================] - ETA: 0s - loss: 1.2103 -
acc: 0.5409
Epoch 00044: val_acc did not improve from 0.52590
72/72 [==============================] - 72s 994ms/step - loss: 1.
2103 - acc: 0.5409 - val_loss: 1.2305 - val_acc: 0.5060
Epoch 45/80
72/72 [==============================] - ETA: 0s - loss: 1.1233 -
acc: 0.5611
Epoch 00045: val_acc did not improve from 0.52590
72/72 [==============================] - 71s 989ms/step - loss: 1.
1233 - acc: 0.5611 - val_loss: 1.1601 - val_acc: 0.5139
Epoch 46/80
72/72 [==============================] - ETA: 0s - loss: 1.1851 -
acc: 0.5356
Epoch 00046: val_acc did not improve from 0.52590
72/72 [==============================] - 70s 979ms/step - loss: 1.
1851 - acc: 0.5356 - val_loss: 1.1923 - val_acc: 0.5219
Epoch 47/80
72/72 [==============================] - ETA: 0s - loss: 1.3626 -
acc: 0.4662
Epoch 00047: val_acc did not improve from 0.52590
72/72 [==============================] - 71s 993ms/step - loss: 1.
3626 - acc: 0.4662 - val_loss: 1.2095 - val_acc: 0.5100
Epoch 48/80
72/72 [==============================] - ETA: 0s - loss: 1.2365 -
acc: 0.5114
Epoch 00048: val_acc did not improve from 0.52590
```

```
72/72 [==============================] - 72s 1s/step - loss: 1.236
5 - acc: 0.5114 - val_loss: 1.2456 - val_acc: 0.5100
Epoch 49/80
72/72 [==============================] - ETA: 0s - loss: 1.1484 -
acc: 0.5510
Epoch 00049: val_acc improved from 0.52590 to 0.54183, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 72s 994ms/step - loss: 1.
1484 - acc: 0.5510 - val_loss: 1.1692 - val_acc: 0.5418
Epoch 50/80
72/72 [==============================] - ETA: 0s - loss: 1.1194 -
acc: 0.5650
Epoch 00050: val_acc did not improve from 0.54183
72/72 [==============================] - 71s 990ms/step - loss: 1.
1194 - acc: 0.5650 - val_loss: 1.2132 - val_acc: 0.5100
Epoch 51/80
72/72 [==============================] - ETA: 0s - loss: 1.1369 -
acc: 0.5540
Epoch 00051: val_acc improved from 0.54183 to 0.54582, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 74s 1s/step - loss: 1.136
9 - acc: 0.5540 - val_loss: 1.1945 - val_acc: 0.5458
Epoch 52/80
72/72 [==============================] - ETA: 0s - loss: 1.1062 -
acc: 0.5712
Epoch 00052: val_acc did not improve from 0.54582
72/72 [==============================] - 76s 1s/step - loss: 1.106
2 - acc: 0.5712 - val_loss: 1.2262 - val_acc: 0.5020
Epoch 53/80
72/72 [==============================] - ETA: 0s - loss: 1.0577 -
acc: 0.5879
Epoch 00053: val_acc did not improve from 0.54582
72/72 [==============================] - 79s 1s/step - loss: 1.057
7 - acc: 0.5879 - val_loss: 1.1375 - val_acc: 0.5299
Epoch 54/80
71/72 [============================>.] - ETA: 1s - loss: 1.0482 -
acc: 0.5849
Epoch 00054: val_acc did not improve from 0.54582
72/72 [==============================] - 80s 1s/step - loss: 1.051
9 - acc: 0.5848 - val_loss: 1.1744 - val_acc: 0.5259
Epoch 55/80
72/72 [==============================] - ETA: 0s - loss: 1.0675 -
acc: 0.5844
Epoch 00055: val_acc did not improve from 0.54582
72/72 [==============================] - 83s 1s/step - loss: 1.067
5 - acc: 0.5844 - val_loss: 1.1345 - val_acc: 0.5418
Epoch 56/80
72/72 [==============================] - ETA: 0s - loss: 1.0591 -
acc: 0.5923
Epoch 00056: val_acc improved from 0.54582 to 0.56175, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 84s 1s/step - loss: 1.059
1 - acc: 0.5923 - val_loss: 1.0868 - val_acc: 0.5618
Epoch 57/80
72/72 [==============================] - ETA: 0s - loss: 1.1965 -
acc: 0.5088
Epoch 00057: val_acc did not improve from 0.56175
```

```
72/72 [==============================] - 85s 1s/step - loss: 1.196
5 - acc: 0.5088 - val_loss: 1.1695 - val_acc: 0.5299
Epoch 58/80
72/72 [==============================] - ETA: 0s - loss: 1.1005 -
acc: 0.5558
Epoch 00058: val_acc did not improve from 0.56175
72/72 [==============================] - 85s 1s/step - loss: 1.100
5 - acc: 0.5558 - val_loss: 1.1129 - val_acc: 0.5458
Epoch 59/80
72/72 [==============================] - ETA: 0s - loss: 1.0267 -
acc: 0.5888
Epoch 00060: val_acc did not improve from 0.56175
72/72 [==============================] - 83s 1s/step - loss: 1.026
7 - acc: 0.5888 - val_loss: 1.1911 - val_acc: 0.5458
Epoch 61/80
72/72 [==============================] - ETA: 0s - loss: 1.0764 -
acc: 0.5694
Epoch 00061: val_acc did not improve from 0.56175
72/72 [==============================] - 80s 1s/step - loss: 1.076
4 - acc: 0.5694 - val_loss: 1.1433 - val_acc: 0.5219
Epoch 62/80
72/72 [==============================] - ETA: 0s - loss: 1.0176 -
acc: 0.5914
Epoch 00062: val_acc did not improve from 0.56175
72/72 [==============================] - 81s 1s/step - loss: 1.017
6 - acc: 0.5914 - val_loss: 1.1927 - val_acc: 0.5418
Epoch 63/80
72/72 [==============================] - ETA: 0s - loss: 1.0087 -
acc: 0.6103
Epoch 00063: val_acc did not improve from 0.56175
72/72 [==============================] - 80s 1s/step - loss: 1.008
7 - acc: 0.6103 - val_loss: 1.2377 - val_acc: 0.5219
Epoch 64/80
72/72 [==============================] - ETA: 0s - loss: 1.0750 -
acc: 0.5773
Epoch 00064: val_acc did not improve from 0.56175
72/72 [==============================] - 77s 1s/step - loss: 1.075
0 - acc: 0.5773 - val_loss: 1.1754 - val_acc: 0.5538
Epoch 65/80
72/72 [==============================] - ETA: 0s - loss: 0.9867 -
acc: 0.6125- ETA: 9s - loss: 0.9992 - acc
Epoch 00065: val_acc improved from 0.56175 to 0.58167, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 76s 1s/step - loss: 0.986
7 - acc: 0.6125 - val_loss: 1.0773 - val_acc: 0.5817
Epoch 66/80
72/72 [==============================] - ETA: 0s - loss: 1.0083 -
acc: 0.6195
Epoch 00066: val_acc did not improve from 0.58167
72/72 [==============================] - 76s 1s/step - loss: 1.008
3 - acc: 0.6195 - val_loss: 1.1478 - val_acc: 0.5378
Epoch 67/80
72/72 [==============================] - ETA: 0s - loss: 0.9965 -
acc: 0.6164
Epoch 00067: val_acc did not improve from 0.58167
72/72 [==============================] - 76s 1s/step - loss: 0.996
5 - acc: 0.6164 - val_loss: 1.1813 - val_acc: 0.5179
```

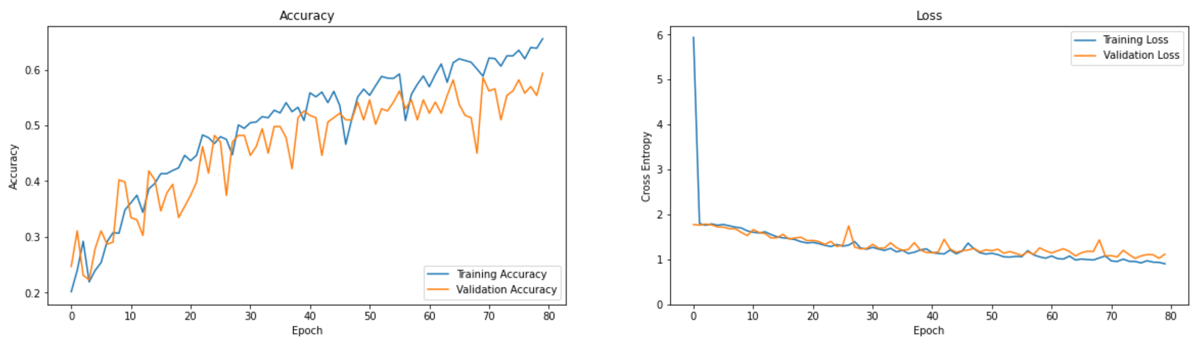
```
Epoch 68/80
72/72 [==============================] - ETA: 0s - loss: 0.9874 -
acc: 0.6134
Epoch 00068: val_acc did not improve from 0.58167
72/72 [==============================] - 76s 1s/step - loss: 0.987
4 - acc: 0.6134 - val_loss: 1.1777 - val_acc: 0.5139
Epoch 69/80
72/72 [==============================] - ETA: 0s - loss: 1.0334 -
acc: 0.6011
Epoch 00069: val_acc did not improve from 0.58167
72/72 [==============================] - 74s 1s/step - loss: 1.033
4 - acc: 0.6011 - val_loss: 1.4337 - val_acc: 0.4502
Epoch 70/80
72/72 [==============================] - ETA: 0s - loss: 1.0770 -
acc: 0.5883
Epoch 00070: val_acc improved from 0.58167 to 0.58566, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 74s 1s/step - loss: 1.077
0 - acc: 0.5883 - val_loss: 1.0788 - val_acc: 0.5857
Epoch 71/80
72/72 [==============================] - ETA: 0s - loss: 0.9674 -
acc: 0.6208
Epoch 00071: val_acc did not improve from 0.58566
72/72 [==============================] - 73s 1s/step - loss: 0.967
4 - acc: 0.6208 - val_loss: 1.0825 - val_acc: 0.5618
Epoch 72/80
71/72 [============================>.] - ETA: 0s - loss: 0.9549 -
acc: 0.6193
Epoch 00072: val_acc did not improve from 0.58566
72/72 [==============================] - 74s 1s/step - loss: 0.953
6 - acc: 0.6199 - val_loss: 1.0552 - val_acc: 0.5657
Epoch 73/80
72/72 [==============================] - ETA: 0s - loss: 1.0069 -
acc: 0.6063
Epoch 00073: val_acc did not improve from 0.58566
72/72 [==============================] - 72s 1s/step - loss: 1.006
9 - acc: 0.6063 - val_loss: 1.2029 - val_acc: 0.5100
Epoch 74/80
72/72 [==============================] - ETA: 0s - loss: 0.9582 -
acc: 0.6248
Epoch 00074: val_acc did not improve from 0.58566
72/72 [==============================] - 74s 1s/step - loss: 0.958
2 - acc: 0.6248 - val_loss: 1.1073 - val_acc: 0.5538
Epoch 75/80
72/72 [==============================] - ETA: 0s - loss: 0.9536 -
acc: 0.6248
Epoch 00075: val_acc did not improve from 0.58566
72/72 [==============================] - 72s 1s/step - loss: 0.953
6 - acc: 0.6248 - val_loss: 1.0219 - val_acc: 0.5618
Epoch 76/80
72/72 [==============================] - ETA: 0s - loss: 0.9224 -
acc: 0.6349
Epoch 00076: val_acc did not improve from 0.58566
72/72 [==============================] - 72s 994ms/step - loss: 0.
9224 - acc: 0.6349 - val_loss: 1.0790 - val_acc: 0.5817
Epoch 77/80
72/72 [==============================] - ETA: 0s - loss: 0.9703 -
```

```
acc: 0.6195
Epoch 00077: val_acc did not improve from 0.58566
72/72 [==============================] - 71s 991ms/step - loss: 0.
9703 - acc: 0.6195 - val_loss: 1.1078 - val_acc: 0.5578
Epoch 78/80
72/72 [==============================] - ETA: 0s - loss: 0.9387 -
acc: 0.6397
Epoch 00078: val_acc did not improve from 0.58566
72/72 [==============================] - 72s 1s/step - loss: 0.938
7 - acc: 0.6397 - val_loss: 1.1041 - val_acc: 0.5697
Epoch 79/80
72/72 [==============================] - ETA: 0s - loss: 0.9346 -
acc: 0.6384
Epoch 00079: val_acc did not improve from 0.58566
72/72 [==============================] - 72s 1s/step - loss: 0.934
6 - acc: 0.6384 - val_loss: 1.0255 - val_acc: 0.5538
Epoch 80/80
72/72 [==============================] - ETA: 0s - loss: 0.9025 -
acc: 0.6555
Epoch 00080: val_acc improved from 0.58566 to 0.59363, saving mode
l to best_cnn_model.h5
72/72 [==============================] - 72s 995ms/step - loss: 0.
9025 - acc: 0.6555 - val_loss: 1.1178 - val_acc: 0.5936
```

Results analysis: Convolutional model.

In order to get consistent results we evaluate the performance of the convolutional model with use the same metrics as the logistic regression model. So we plot accuracy and loss and calculate confusion matrix and F1 score.

In [24]:
```
plot_history(cnn_history)
```

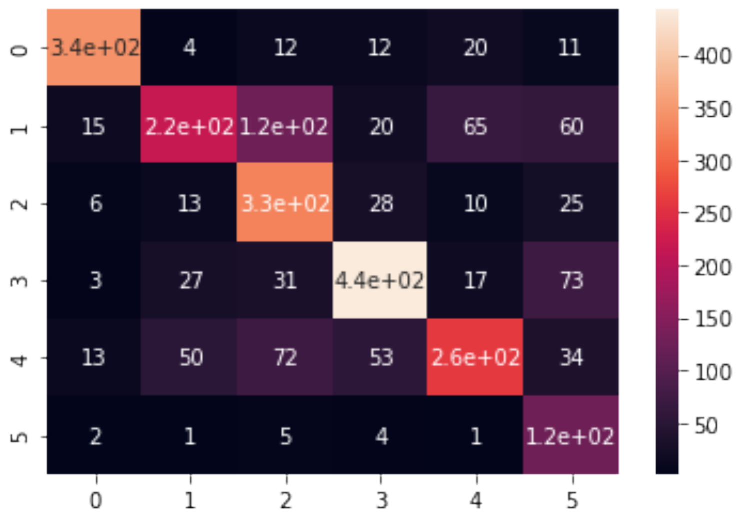


Let's make the predictions

In [33]:
```
cnn_ys, cnn_preds = evaluate(cnn, test_generator)
```

Confusion matrix

```
In [34]: confusion(cnn_ys, cnn_preds)
```



F1 score

```
In [35]: f1_cnn = f1(cnn_ys, cnn_preds)
```

```
F1 score is: 0.6792687080815812
```

Train the model with data normalization.

In order to get better results a good strategy is to normalize the input. In this case we restrict all values in the interval $[0, 1]$. Having similar distribution between samples ensures a faster convergence during training and avoids the vanishing gradient problem, which can slow down or halt the training process. By limiting values the gradient also remains bounded, thus avoiding also exploding gradient. We perform a new training on the Convolutional Model with normalized data and then compare the results.

In [20]:
```
tr.rescale = 1./255
val.rescale = 1./255

train_norm_generator = tr.flow_from_directory(IMG_DIR,
                                              target_size=img_size,
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              subset='training')

val_norm_generator=val.flow_from_directory(IMG_DIR,
                                            target_size=img_size,
                                            batch_size=batch_size,
                                            class_mode='categorical',
                                            subset='validation')

test_norm_generator = ImageDataGenerator(rescale=1./255).flow_from_
directory(IMG_DIR,
                                            target_size=img_size,
                                            batch_size=1,
                                            class_mode='categorical')
```

```
Found 2276 images belonging to 6 classes.
Found 251 images belonging to 6 classes.
Found 2527 images belonging to 6 classes.
```

In [22]:
```
cnn_norm = CNN_model()
cnn_norm_history = train(cnn_norm, train_norm_generator, val_norm_g
enerator, epochs = 80, best_model_name='best_cnn_model_norm.h5')
```

```
Epoch 1/80
72/72 [==============================] - ETA: 0s - loss: 1.7758 - 
acc: 0.2272
Epoch 00001: val_acc improved from -inf to 0.29084, saving model t
o best_cnn_model_norm.h5
72/72 [==============================] - 84s 1s/step - loss: 1.775
8 - acc: 0.2272 - val_loss: 1.6933 - val_acc: 0.2908
Epoch 2/80
72/72 [==============================] - ETA: 0s - loss: 1.6869 - 
acc: 0.2935
Epoch 00002: val_acc improved from 0.29084 to 0.35060, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 82s 1s/step - loss: 1.686
9 - acc: 0.2935 - val_loss: 1.6204 - val_acc: 0.3506
Epoch 3/80
72/72 [==============================] - ETA: 0s - loss: 1.5657 - 
acc: 0.3572
Epoch 00003: val_acc improved from 0.35060 to 0.38645, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 84s 1s/step - loss: 1.565
7 - acc: 0.3572 - val_loss: 1.4723 - val_acc: 0.3865
Epoch 4/80
72/72 [==============================] - ETA: 0s - loss: 1.4932 - 
acc: 0.4165
Epoch 00004: val_acc did not improve from 0.38645
72/72 [==============================] - 83s 1s/step - loss: 1.493
2 - acc: 0.4165 - val_loss: 1.4910 - val_acc: 0.3825
```

```
Epoch 5/80
72/72 [==============================] - ETA: 0s - loss: 1.4713 -
acc: 0.4029
Epoch 00005: val_acc improved from 0.38645 to 0.43028, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 84s 1s/step - loss: 1.471
3 - acc: 0.4029 - val_loss: 1.3936 - val_acc: 0.4303
Epoch 6/80
72/72 [==============================] - ETA: 0s - loss: 1.3584 -
acc: 0.4583
Epoch 00006: val_acc improved from 0.43028 to 0.50598, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 84s 1s/step - loss: 1.358
4 - acc: 0.4583 - val_loss: 1.2880 - val_acc: 0.5060
Epoch 7/80
72/72 [==============================] - ETA: 0s - loss: 1.3557 -
acc: 0.4750
Epoch 00007: val_acc did not improve from 0.50598
72/72 [==============================] - 85s 1s/step - loss: 1.355
7 - acc: 0.4750 - val_loss: 1.4108 - val_acc: 0.4582
Epoch 8/80
72/72 [==============================] - ETA: 0s - loss: 1.2880 -
acc: 0.5009
Epoch 00008: val_acc did not improve from 0.50598
72/72 [==============================] - 83s 1s/step - loss: 1.288
0 - acc: 0.5009 - val_loss: 1.3106 - val_acc: 0.4701
Epoch 9/80
72/72 [==============================] - ETA: 0s - loss: 1.2897 -
acc: 0.4956
Epoch 00009: val_acc improved from 0.50598 to 0.52988, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 85s 1s/step - loss: 1.289
7 - acc: 0.4956 - val_loss: 1.2211 - val_acc: 0.5299
Epoch 10/80
72/72 [==============================] - ETA: 0s - loss: 1.1826 -
acc: 0.5492
Epoch 00010: val_acc did not improve from 0.52988
72/72 [==============================] - 84s 1s/step - loss: 1.182
6 - acc: 0.5492 - val_loss: 1.2971 - val_acc: 0.4900
Epoch 11/80
72/72 [==============================] - ETA: 0s - loss: 1.2691 -
acc: 0.5176
Epoch 00011: val_acc did not improve from 0.52988
72/72 [==============================] - 85s 1s/step - loss: 1.269
1 - acc: 0.5176 - val_loss: 1.2271 - val_acc: 0.4980
Epoch 12/80
72/72 [==============================] - ETA: 0s - loss: 1.1755 -
acc: 0.5501
Epoch 00012: val_acc improved from 0.52988 to 0.54183, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 85s 1s/step - loss: 1.175
5 - acc: 0.5501 - val_loss: 1.1255 - val_acc: 0.5418
Epoch 13/80
72/72 [==============================] - ETA: 0s - loss: 1.1503 -
acc: 0.5461
Epoch 00013: val_acc did not improve from 0.54183
72/72 [==============================] - 86s 1s/step - loss: 1.150
```

```
3 - acc: 0.5461 - val_loss: 1.3567 - val_acc: 0.4980
Epoch 14/80
72/72 [==============================] - ETA: 0s - loss: 1.1630 -
acc: 0.5479
Epoch 00014: val_acc did not improve from 0.54183
72/72 [==============================] - 84s 1s/step - loss: 1.163
0 - acc: 0.5479 - val_loss: 1.2345 - val_acc: 0.5259
Epoch 15/80
72/72 [==============================] - ETA: 0s - loss: 1.1689 -
acc: 0.5571
Epoch 00015: val_acc did not improve from 0.54183
72/72 [==============================] - 86s 1s/step - loss: 1.168
9 - acc: 0.5571 - val_loss: 1.2486 - val_acc: 0.5219
Epoch 16/80
72/72 [==============================] - ETA: 0s - loss: 1.1196 -
acc: 0.5756
Epoch 00016: val_acc did not improve from 0.54183
72/72 [==============================] - 87s 1s/step - loss: 1.119
6 - acc: 0.5756 - val_loss: 1.1781 - val_acc: 0.5339
Epoch 17/80
72/72 [==============================] - ETA: 0s - loss: 1.0860 -
acc: 0.5734
Epoch 00017: val_acc improved from 0.54183 to 0.54582, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 86s 1s/step - loss: 1.086
0 - acc: 0.5734 - val_loss: 1.3092 - val_acc: 0.5458
Epoch 18/80
72/72 [==============================] - ETA: 0s - loss: 1.0811 -
acc: 0.5822
Epoch 00018: val_acc improved from 0.54582 to 0.59761, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 88s 1s/step - loss: 1.081
1 - acc: 0.5822 - val_loss: 1.0256 - val_acc: 0.5976
Epoch 19/80
72/72 [==============================] - ETA: 0s - loss: 1.0655 -
acc: 0.6015
Epoch 00019: val_acc did not improve from 0.59761
72/72 [==============================] - 86s 1s/step - loss: 1.065
5 - acc: 0.6015 - val_loss: 1.1506 - val_acc: 0.5418
Epoch 20/80
72/72 [==============================] - ETA: 0s - loss: 1.0435 -
acc: 0.5936
Epoch 00020: val_acc did not improve from 0.59761
72/72 [==============================] - 87s 1s/step - loss: 1.043
5 - acc: 0.5936 - val_loss: 1.0947 - val_acc: 0.5817
Epoch 21/80
72/72 [==============================] - ETA: 0s - loss: 0.9854 -
acc: 0.6239
Epoch 00021: val_acc improved from 0.59761 to 0.60558, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 87s 1s/step - loss: 0.985
4 - acc: 0.6239 - val_loss: 1.0794 - val_acc: 0.6056
Epoch 22/80
72/72 [==============================] - ETA: 0s - loss: 1.0338 -
acc: 0.6072
Epoch 00022: val_acc did not improve from 0.60558
72/72 [==============================] - 88s 1s/step - loss: 1.033
```

```
8 - acc: 0.6072 - val_loss: 1.1299 - val_acc: 0.5657
Epoch 23/80
72/72 [==============================] - ETA: 0s - loss: 0.9812 -
acc: 0.6344
Epoch 00023: val_acc did not improve from 0.60558
72/72 [==============================] - 87s 1s/step - loss: 0.981
2 - acc: 0.6344 - val_loss: 1.1273 - val_acc: 0.5458
Epoch 24/80
72/72 [==============================] - ETA: 0s - loss: 0.9876 -
acc: 0.6397
Epoch 00024: val_acc did not improve from 0.60558
72/72 [==============================] - 87s 1s/step - loss: 0.987
6 - acc: 0.6397 - val_loss: 1.0631 - val_acc: 0.5737
Epoch 25/80
72/72 [==============================] - ETA: 0s - loss: 0.9522 -
acc: 0.6507
Epoch 00025: val_acc did not improve from 0.60558
72/72 [==============================] - 88s 1s/step - loss: 0.952
2 - acc: 0.6507 - val_loss: 1.0819 - val_acc: 0.5777
Epoch 26/80
72/72 [==============================] - ETA: 0s - loss: 0.9764 -
acc: 0.6437
Epoch 00026: val_acc did not improve from 0.60558
72/72 [==============================] - 87s 1s/step - loss: 0.976
4 - acc: 0.6437 - val_loss: 1.1131 - val_acc: 0.5697
Epoch 27/80
72/72 [==============================] - ETA: 0s - loss: 0.9380 -
acc: 0.6454
Epoch 00027: val_acc improved from 0.60558 to 0.62550, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 88s 1s/step - loss: 0.938
0 - acc: 0.6454 - val_loss: 0.9818 - val_acc: 0.6255
Epoch 28/80
72/72 [==============================] - ETA: 0s - loss: 0.8873 -
acc: 0.6586
Epoch 00028: val_acc did not improve from 0.62550
72/72 [==============================] - 87s 1s/step - loss: 0.887
3 - acc: 0.6586 - val_loss: 0.9643 - val_acc: 0.6016
Epoch 29/80
72/72 [==============================] - ETA: 0s - loss: 0.9502 -
acc: 0.6406
Epoch 00029: val_acc did not improve from 0.62550
72/72 [==============================] - 86s 1s/step - loss: 0.950
2 - acc: 0.6406 - val_loss: 0.9271 - val_acc: 0.6175
Epoch 30/80
72/72 [==============================] - ETA: 0s - loss: 0.8961 -
acc: 0.6533
Epoch 00030: val_acc improved from 0.62550 to 0.65737, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 87s 1s/step - loss: 0.896
1 - acc: 0.6533 - val_loss: 0.9748 - val_acc: 0.6574
Epoch 31/80
72/72 [==============================] - ETA: 0s - loss: 0.9426 -
acc: 0.6529
Epoch 00031: val_acc did not improve from 0.65737
72/72 [==============================] - 87s 1s/step - loss: 0.942
6 - acc: 0.6529 - val_loss: 1.0255 - val_acc: 0.6056
```

```
Epoch 32/80
72/72 [==============================] - ETA: 0s - loss: 0.8754 -
acc: 0.6709
Epoch 00032: val_acc did not improve from 0.65737
72/72 [==============================] - 88s 1s/step - loss: 0.875
4 - acc: 0.6709 - val_loss: 1.0610 - val_acc: 0.5936
Epoch 33/80
72/72 [==============================] - ETA: 0s - loss: 0.8544 -
acc: 0.6863
Epoch 00033: val_acc improved from 0.65737 to 0.66135, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 87s 1s/step - loss: 0.854
4 - acc: 0.6863 - val_loss: 0.8511 - val_acc: 0.6614
Epoch 34/80
72/72 [==============================] - ETA: 0s - loss: 0.8379 -
acc: 0.6863
Epoch 00034: val_acc did not improve from 0.66135
72/72 [==============================] - 88s 1s/step - loss: 0.837
9 - acc: 0.6863 - val_loss: 0.8858 - val_acc: 0.6614
Epoch 35/80
72/72 [==============================] - ETA: 0s - loss: 0.8237 -
acc: 0.6872
Epoch 00035: val_acc did not improve from 0.66135
72/72 [==============================] - 88s 1s/step - loss: 0.823
7 - acc: 0.6872 - val_loss: 1.0534 - val_acc: 0.6295
Epoch 36/80
72/72 [==============================] - ETA: 0s - loss: 0.8227 -
acc: 0.6986
Epoch 00036: val_acc did not improve from 0.66135
72/72 [==============================] - 87s 1s/step - loss: 0.822
7 - acc: 0.6986 - val_loss: 0.9530 - val_acc: 0.6454
Epoch 37/80
72/72 [==============================] - ETA: 0s - loss: 0.8446 -
acc: 0.6832
Epoch 00037: val_acc did not improve from 0.66135
72/72 [==============================] - 88s 1s/step - loss: 0.844
6 - acc: 0.6832 - val_loss: 0.9861 - val_acc: 0.6295
Epoch 38/80
72/72 [==============================] - ETA: 0s - loss: 0.7926 -
acc: 0.7039
Epoch 00038: val_acc did not improve from 0.66135
72/72 [==============================] - 87s 1s/step - loss: 0.792
6 - acc: 0.7039 - val_loss: 1.1780 - val_acc: 0.5618
Epoch 39/80
72/72 [==============================] - ETA: 0s - loss: 0.8801 -
acc: 0.6736
Epoch 00039: val_acc did not improve from 0.66135
72/72 [==============================] - 88s 1s/step - loss: 0.880
1 - acc: 0.6736 - val_loss: 1.1155 - val_acc: 0.5896
Epoch 40/80
72/72 [==============================] - ETA: 0s - loss: 0.8005 -
acc: 0.6863
Epoch 00040: val_acc did not improve from 0.66135
72/72 [==============================] - 86s 1s/step - loss: 0.800
5 - acc: 0.6863 - val_loss: 0.8844 - val_acc: 0.6574
Epoch 41/80
72/72 [==============================] - ETA: 0s - loss: 0.7776 -
```

```
acc: 0.7210
Epoch 00041: val_acc did not improve from 0.66135
72/72 [==============================] - 86s 1s/step - loss: 0.777
6 - acc: 0.7210 - val_loss: 0.9806 - val_acc: 0.6454
Epoch 42/80
72/72 [==============================] - ETA: 0s - loss: 0.7526 -
acc: 0.7179
Epoch 00042: val_acc did not improve from 0.66135
72/72 [==============================] - 88s 1s/step - loss: 0.752
6 - acc: 0.7179 - val_loss: 0.9714 - val_acc: 0.6454
Epoch 43/80
72/72 [==============================] - ETA: 0s - loss: 0.7942 -
acc: 0.7065
Epoch 00043: val_acc did not improve from 0.66135
72/72 [==============================] - 86s 1s/step - loss: 0.794
2 - acc: 0.7065 - val_loss: 0.8987 - val_acc: 0.6494
Epoch 44/80
72/72 [==============================] - ETA: 0s - loss: 0.6576 -
acc: 0.7487
Epoch 00044: val_acc improved from 0.66135 to 0.67729, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 89s 1s/step - loss: 0.657
6 - acc: 0.7487 - val_loss: 0.8319 - val_acc: 0.6773
Epoch 45/80
72/72 [==============================] - ETA: 0s - loss: 0.6240 -
acc: 0.7676
Epoch 00045: val_acc did not improve from 0.67729
72/72 [==============================] - 88s 1s/step - loss: 0.624
0 - acc: 0.7676 - val_loss: 0.8373 - val_acc: 0.6733
Epoch 46/80
72/72 [==============================] - ETA: 0s - loss: 0.6025 -
acc: 0.7715
Epoch 00046: val_acc improved from 0.67729 to 0.70120, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 88s 1s/step - loss: 0.602
5 - acc: 0.7715 - val_loss: 0.8170 - val_acc: 0.7012
Epoch 47/80
72/72 [==============================] - ETA: 0s - loss: 0.6030 -
acc: 0.7733
Epoch 00047: val_acc did not improve from 0.70120
72/72 [==============================] - 89s 1s/step - loss: 0.603
0 - acc: 0.7733 - val_loss: 0.7938 - val_acc: 0.6932
Epoch 48/80
72/72 [==============================] - ETA: 0s - loss: 0.5893 -
acc: 0.7878
Epoch 00048: val_acc did not improve from 0.70120
72/72 [==============================] - 88s 1s/step - loss: 0.589
3 - acc: 0.7878 - val_loss: 0.8439 - val_acc: 0.6813
Epoch 49/80
72/72 [==============================] - ETA: 0s - loss: 0.5959 -
acc: 0.7768
Epoch 00049: val_acc did not improve from 0.70120
72/72 [==============================] - 89s 1s/step - loss: 0.595
9 - acc: 0.7768 - val_loss: 0.8153 - val_acc: 0.6853
Epoch 50/80
72/72 [==============================] - ETA: 0s - loss: 0.6020 -
acc: 0.7755
```

```
Epoch 00050: val_acc did not improve from 0.70120
72/72 [==============================] - 88s 1s/step - loss: 0.602
0 - acc: 0.7755 - val_loss: 0.8033 - val_acc: 0.6932
Epoch 51/80
72/72 [==============================] - ETA: 0s - loss: 0.6017 -
acc: 0.7777
Epoch 00051: val_acc improved from 0.70120 to 0.70518, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 89s 1s/step - loss: 0.601
7 - acc: 0.7777 - val_loss: 0.8211 - val_acc: 0.7052
Epoch 52/80
72/72 [==============================] - ETA: 0s - loss: 0.5613 -
acc: 0.7957
Epoch 00052: val_acc did not improve from 0.70518
72/72 [==============================] - 88s 1s/step - loss: 0.561
3 - acc: 0.7957 - val_loss: 0.8152 - val_acc: 0.7012
Epoch 53/80
72/72 [==============================] - ETA: 0s - loss: 0.5754 -
acc: 0.7926
Epoch 00053: val_acc did not improve from 0.70518
72/72 [==============================] - 87s 1s/step - loss: 0.575
4 - acc: 0.7926 - val_loss: 0.8459 - val_acc: 0.7012
Epoch 54/80
72/72 [==============================] - ETA: 0s - loss: 0.5698 -
acc: 0.7843
Epoch 00054: val_acc did not improve from 0.70518
72/72 [==============================] - 88s 1s/step - loss: 0.569
8 - acc: 0.7843 - val_loss: 0.8371 - val_acc: 0.6853
Epoch 55/80
72/72 [==============================] - ETA: 0s - loss: 0.5959 -
acc: 0.7803
Epoch 00055: val_acc did not improve from 0.70518
72/72 [==============================] - 88s 1s/step - loss: 0.595
9 - acc: 0.7803 - val_loss: 0.8001 - val_acc: 0.7012
Epoch 56/80
72/72 [==============================] - ETA: 0s - loss: 0.5776 -
acc: 0.7838
Epoch 00056: val_acc improved from 0.70518 to 0.70916, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 88s 1s/step - loss: 0.577
6 - acc: 0.7838 - val_loss: 0.7927 - val_acc: 0.7092
Epoch 57/80
72/72 [==============================] - ETA: 0s - loss: 0.5762 -
acc: 0.7869
Epoch 00057: val_acc did not improve from 0.70916
72/72 [==============================] - 89s 1s/step - loss: 0.576
2 - acc: 0.7869 - val_loss: 0.8129 - val_acc: 0.7012
Epoch 58/80
72/72 [==============================] - ETA: 0s - loss: 0.5604 -
acc: 0.7860
Epoch 00058: val_acc did not improve from 0.70916
72/72 [==============================] - 88s 1s/step - loss: 0.560
4 - acc: 0.7860 - val_loss: 0.8259 - val_acc: 0.7012
Epoch 59/80
72/72 [==============================] - ETA: 0s - loss: 0.5490 -
acc: 0.7913
Epoch 00059: val_acc did not improve from 0.70916
```

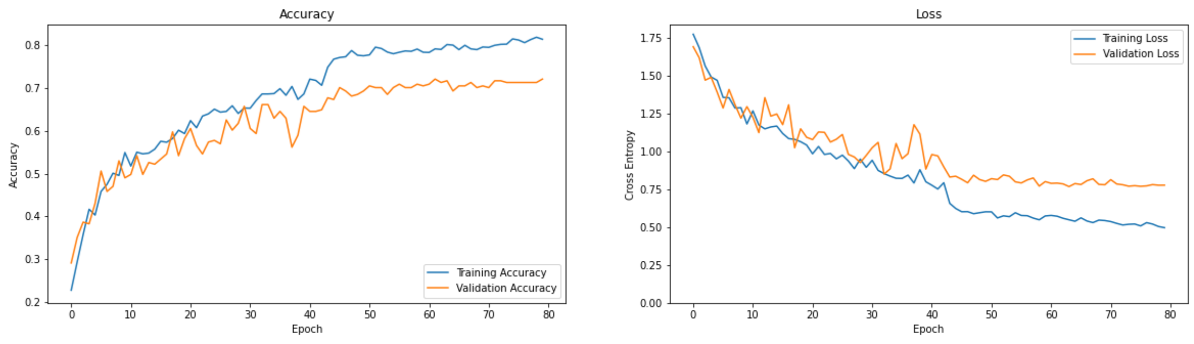
```
72/72 [==============================] - 89s 1s/step - loss: 0.549
0 - acc: 0.7913 - val_loss: 0.7719 - val_acc: 0.7092
Epoch 60/80
72/72 [==============================] - ETA: 0s - loss: 0.5745 -
acc: 0.7838
Epoch 00060: val_acc did not improve from 0.70916
72/72 [==============================] - 89s 1s/step - loss: 0.574
5 - acc: 0.7838 - val_loss: 0.8018 - val_acc: 0.7052
Epoch 61/80
72/72 [==============================] - ETA: 0s - loss: 0.5781 -
acc: 0.7834
Epoch 00061: val_acc did not improve from 0.70916
72/72 [==============================] - 89s 1s/step - loss: 0.578
1 - acc: 0.7834 - val_loss: 0.7902 - val_acc: 0.7092
Epoch 62/80
72/72 [==============================] - ETA: 0s - loss: 0.5729 -
acc: 0.7917
Epoch 00062: val_acc improved from 0.70916 to 0.72112, saving mode
l to best_cnn_model_norm.h5
72/72 [==============================] - 90s 1s/step - loss: 0.572
9 - acc: 0.7917 - val_loss: 0.7919 - val_acc: 0.7211
Epoch 63/80
72/72 [==============================] - ETA: 0s - loss: 0.5591 -
acc: 0.7904
Epoch 00063: val_acc did not improve from 0.72112
72/72 [==============================] - 88s 1s/step - loss: 0.559
1 - acc: 0.7904 - val_loss: 0.7864 - val_acc: 0.7131
Epoch 64/80
72/72 [==============================] - ETA: 0s - loss: 0.5492 -
acc: 0.8018
Epoch 00064: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.549
2 - acc: 0.8018 - val_loss: 0.7688 - val_acc: 0.7171
Epoch 65/80
72/72 [==============================] - ETA: 0s - loss: 0.5395 -
acc: 0.8005
Epoch 00065: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.539
5 - acc: 0.8005 - val_loss: 0.7888 - val_acc: 0.6932
Epoch 66/80
72/72 [==============================] - ETA: 0s - loss: 0.5621 -
acc: 0.7900
Epoch 00066: val_acc did not improve from 0.72112
72/72 [==============================] - 88s 1s/step - loss: 0.562
1 - acc: 0.7900 - val_loss: 0.7824 - val_acc: 0.7052
Epoch 67/80
72/72 [==============================] - ETA: 0s - loss: 0.5413 -
acc: 0.8001
Epoch 00067: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.541
3 - acc: 0.8001 - val_loss: 0.8074 - val_acc: 0.7052
Epoch 68/80
72/72 [==============================] - ETA: 0s - loss: 0.5306 -
acc: 0.7917
Epoch 00068: val_acc did not improve from 0.72112
72/72 [==============================] - 87s 1s/step - loss: 0.530
6 - acc: 0.7917 - val_loss: 0.8203 - val_acc: 0.7131
```

```
Epoch 69/80
72/72 [==============================] - ETA: 0s - loss: 0.5471 -
acc: 0.7900
Epoch 00069: val_acc did not improve from 0.72112
72/72 [==============================] - 87s 1s/step - loss: 0.547
1 - acc: 0.7900 - val_loss: 0.7833 - val_acc: 0.7012
Epoch 70/80
72/72 [==============================] - ETA: 0s - loss: 0.5442 -
acc: 0.7961
Epoch 00070: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.544
2 - acc: 0.7961 - val_loss: 0.7810 - val_acc: 0.7052
Epoch 71/80
72/72 [==============================] - ETA: 0s - loss: 0.5373 -
acc: 0.7953
Epoch 00071: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.537
3 - acc: 0.7953 - val_loss: 0.8145 - val_acc: 0.7012
Epoch 72/80
72/72 [==============================] - ETA: 0s - loss: 0.5254 -
acc: 0.8001
Epoch 00072: val_acc did not improve from 0.72112
72/72 [==============================] - 90s 1s/step - loss: 0.525
4 - acc: 0.8001 - val_loss: 0.7860 - val_acc: 0.7171
Epoch 73/80
72/72 [==============================] - ETA: 0s - loss: 0.5149 -
acc: 0.8023
Epoch 00073: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.514
9 - acc: 0.8023 - val_loss: 0.7809 - val_acc: 0.7171
Epoch 74/80
72/72 [==============================] - ETA: 0s - loss: 0.5195 -
acc: 0.8027
Epoch 00074: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.519
5 - acc: 0.8027 - val_loss: 0.7714 - val_acc: 0.7131
Epoch 75/80
72/72 [==============================] - ETA: 0s - loss: 0.5214 -
acc: 0.8150
Epoch 00075: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.521
4 - acc: 0.8150 - val_loss: 0.7748 - val_acc: 0.7131
Epoch 76/80
72/72 [==============================] - ETA: 0s - loss: 0.5088 -
acc: 0.8124
Epoch 00076: val_acc did not improve from 0.72112
72/72 [==============================] - 88s 1s/step - loss: 0.508
8 - acc: 0.8124 - val_loss: 0.7707 - val_acc: 0.7131
Epoch 77/80
72/72 [==============================] - ETA: 0s - loss: 0.5304 -
acc: 0.8062
Epoch 00077: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.530
4 - acc: 0.8062 - val_loss: 0.7734 - val_acc: 0.7131
Epoch 78/80
72/72 [==============================] - ETA: 0s - loss: 0.5210 -
acc: 0.8133
```

```
Epoch 00078: val_acc did not improve from 0.72112
72/72 [==============================] - 89s 1s/step - loss: 0.521
0 - acc: 0.8133 - val_loss: 0.7819 - val_acc: 0.7131
Epoch 79/80
72/72 [==============================] - ETA: 0s - loss: 0.5049 -
acc: 0.8190
Epoch 00079: val_acc did not improve from 0.72112
72/72 [==============================] - 88s 1s/step - loss: 0.504
9 - acc: 0.8190 - val_loss: 0.7780 - val_acc: 0.7131
Epoch 80/80
71/72 [============================>.] - ETA: 1s - loss: 0.4977 -
acc: 0.8138
Epoch 00080: val_acc did not improve from 0.72112
72/72 [==============================] - 90s 1s/step - loss: 0.497
1 - acc: 0.8141 - val_loss: 0.7783 - val_acc: 0.7211
```

Results analysis: Convolutional model with normalization

In [23]:
```
plot_history(cnn_norm_history)
```

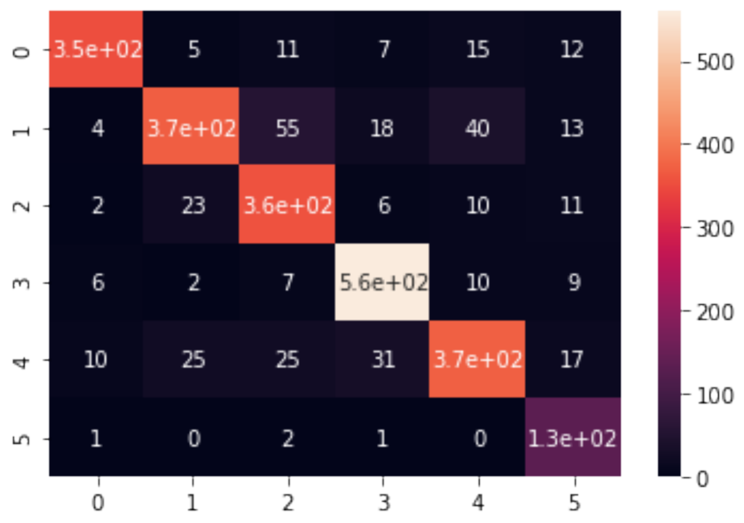


Let's make the predictions

In [24]:
```
cnn_norm_ys, cnn_norm_preds = evaluate(cnn_norm, test_norm_generato
r)
```

Confusion matrix

```
In [26]: confusion(cnn_norm_ys, cnn_norm_preds)
```



F1 Score

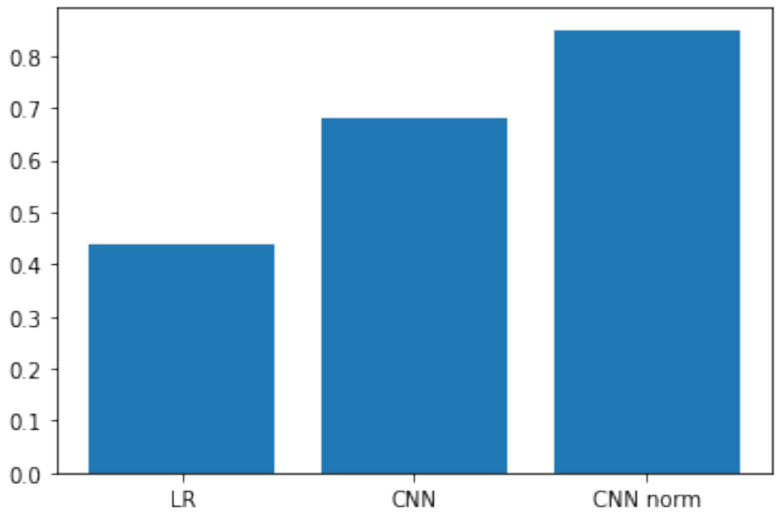
```
In [27]: f1_cnn_norm = f1(cnn_norm_ys, cnn_norm_preds)
```

```
F1 score is: 0.850071763744821
```

Conclusions

Finally let's plot the F1 scores for the three different models

```
In [28]: plt.bar(['LR', 'CNN', 'CNN norm'], [f1_lr, f1_cnn, f1_cnn_norm])
         plt.show()
```



We achieved a final F1 score of 0.85 with the Convolutional model with normalized data. Compared to the Convolutional model with non normalized data shows a 27% improvement and compared to the simple Logistic Regression model a 97% improvement. This shows that feature engineering and model selection are cornerstones of the machine learning workflow and it's important to focus thoroughly and do plenty of testing before deciding.