

```
In [1]: # Convolutional Neural Network

# Installing Theano
# pip install --upgrade --no-deps git+git://github.com/Theano/Theano.git

# Installing Tensorflow
# Install Tensorflow from the website: https://www.tensorflow.org/versions/r0.12/get_started/os_setup.html

# Installing Keras
# pip install --upgrade keras

# Part 1 - Building the CNN

# Importing the Keras Libraries and packages
import numpy as np
import os
import keras_metrics
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import TimeDistributed
from keras.layers import LSTM
from keras.layers import Reshape

import warnings
warnings.filterwarnings('ignore')

# Initialising the CNN
classifier = Sequential()

# Step 1 - Convolution
classifier.add(Convolution2D(64, (3, 3), padding = 'same', input_shape = (128, 128, 3), activation = 'relu'))

# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a second convolutional layer
classifier.add(Convolution2D(64, (3, 3), padding = 'same', activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a third convolutional layer
classifier.add(Convolution2D(64, (3, 3), padding = 'same', activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Step 3 - Flattening
classifier.add(Flatten())
classifier.add(Dropout(rate = 0.5))

# Step 4 - Full connection
```

```

classifier.add(Dense(output_dim = 128, activation = 'relu'))
classifier.add(Dropout(rate = 0.5))
classifier.add(Dense(output_dim = 7, activation = 'softmax'))

```

```

classifier.summary()

```

Z:\Anaconda3\lib\site-packages\h5py\\_\_init\_\_.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.

from .\_conv import register\_converters as \_register\_converters  
Using TensorFlow backend.

Layer (type)	Output Shape	Param #
=====		
conv2d_1 (Conv2D)	(None, 128, 128, 64)	1792
max_pooling2d_1 (MaxPooling2)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 64)	36928
max_pooling2d_2 (MaxPooling2)	(None, 32, 32, 64)	0
conv2d_3 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d_3 (MaxPooling2)	(None, 16, 16, 64)	0
flatten_1 (Flatten)	(None, 16384)	0
dropout_1 (Dropout)	(None, 16384)	0
dense_1 (Dense)	(None, 128)	2097280
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 7)	903
=====		
Total params: 2,173,831		
Trainable params: 2,173,831		
Non-trainable params: 0		
=====		

```

In [2]: # Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy', keras_metrics.precision(), keras_metrics.recall()])

```

```
In [3]: # Part 2 - Fitting the CNN to the images

from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   height_shift_range = 0.1,
                                   width_shift_range = 0.1,
                                   channel_shift_range = 10)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('train/',
                                                target_size = (128, 128),
                                                batch_size = 32,
                                                class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('test/',
                                            target_size = (128, 128),
                                            batch_size = 32,
                                            class_mode = 'categorical')
```

Found 399 images belonging to 7 classes.  
Found 136 images belonging to 7 classes.

```
In [4]: results = classifier.fit_generator(training_set,
                                         samples_per_epoch = 399,
                                         nb_epoch = 100,
                                         validation_data = test_set,
                                         nb_val_samples = 136)
```

Epoch 1/100  
12/12 [=====] - 124s 10s/step - loss: 1.9434 - acc: 0.1994 - precision: 0.0000e+00 - recall: 0.0000e+00 - val\_loss: 1.9024 - val\_acc: 0.2362 - val\_precision: 0.0000e+00 - val\_recall: 0.0000e+00

Epoch 2/100  
12/12 [=====] - 118s 10s/step - loss: 1.8891 - acc: 0.2486 - precision: 0.1673 - recall: 0.0105 - val\_loss: 1.8568 - val\_acc: 0.2352 - val\_precision: 0.0000e+00 - val\_recall: 0.0000e+00

Epoch 3/100  
12/12 [=====] - 120s 10s/step - loss: 1.8104 - acc: 0.2935 - precision: 0.2306 - recall: 0.0184 - val\_loss: 1.6959 - val\_acc: 0.3161 - val\_precision: 0.6366 - val\_recall: 0.0807

Epoch 4/100  
12/12 [=====] - 118s 10s/step - loss: 1.6541 - acc: 0.3796 - precision: 0.6450 - recall: 0.0759 - val\_loss: 1.5186 - val\_acc: 0.3823 - val\_precision: 0.6621 - val\_recall: 0.1180

Epoch 5/100  
12/12 [=====] - 117s 10s/step - loss: 1.6065 - acc: 0.3724 - precision: 0.6375 - recall: 0.1172 - val\_loss: 1.4760 - val\_acc: 0.4563 - val\_precision: 0.8215 - val\_recall: 0.1102

Epoch 6/100  
12/12 [=====] - 116s 10s/step - loss: 1.4788 - acc: 0.4552 - precision: 0.6717 - recall: 0.1842 - val\_loss: 1.3680 - val\_acc: 0.4263 - val\_precision: 0.7068 - val\_recall: 0.2065

Epoch 7/100  
12/12 [=====] - 116s 10s/step - loss: 1.5332 - acc: 0.3923 - precision: 0.6278 - recall: 0.1282 - val\_loss: 1.3909 - val\_acc: 0.4852 - val\_precision: 0.7180 - val\_recall: 0.2203

Epoch 8/100  
12/12 [=====] - 116s 10s/step - loss: 1.4353 - acc: 0.4505 - precision: 0.7143 - recall: 0.1745 - val\_loss: 1.2860 - val\_acc: 0.4560 - val\_precision: 0.6683 - val\_recall: 0.2503

Epoch 9/100  
12/12 [=====] - 118s 10s/step - loss: 1.5372 - acc: 0.4260 - precision: 0.5877 - recall: 0.1078 - val\_loss: 1.2418 - val\_acc: 0.5367 - val\_precision: 0.8353 - val\_recall: 0.1917

Epoch 10/100  
12/12 [=====] - 117s 10s/step - loss: 1.3365 - acc: 0.4688 - precision: 0.6495 - recall: 0.2708 - val\_loss: 1.1758 - val\_acc: 0.6177 - val\_precision: 0.8436 - val\_recall: 0.2055

Epoch 11/100  
12/12 [=====] - 115s 10s/step - loss: 1.3713 - acc: 0.4632 - precision: 0.6881 - recall: 0.1963 - val\_loss: 1.1133 - val\_acc: 0.5805 - val\_precision: 0.8036 - val\_recall: 0.3234

Epoch 12/100  
12/12 [=====] - 118s 10s/step - loss: 1.2984 - acc: 0.5182 - precision: 0.7061 - recall: 0.2537 - val\_loss: 1.1356 - val\_acc: 0.5799 - val\_precision: 0.7814 - val\_recall: 0.2870

Epoch 13/100  
12/12 [=====] - 114s 10s/step - loss: 1.2281 - acc: 0.5290 - precision: 0.6632 - recall: 0.2881 - val\_loss: 1.0175 - val\_acc: 0.5896 - val\_precision: 0.7730 - val\_recall: 0.4271

Epoch 14/100  
12/12 [=====] - 116s 10s/step - loss: 1.1919 - acc: 0.5867 - precision: 0.7118 - recall: 0.3613 - val\_loss: 1.0511 - val\_acc: 0.6180 - val\_precision: 0.7039 - val\_recall: 0.2935

Epoch 15/100

```
12/12 [=====] - 118s 10s/step - loss: 1.3102 - acc:
0.4868 - precision: 0.6465 - recall: 0.2931 - val_loss: 1.0693 - val_acc: 0.6
617 - val_precision: 0.8251 - val_recall: 0.3167
Epoch 16/100
12/12 [=====] - 113s 9s/step - loss: 1.2318 - acc:
0.5183 - precision: 0.7042 - recall: 0.2803 - val_loss: 1.0413 - val_acc: 0.5
886 - val_precision: 0.7504 - val_recall: 0.3971
Epoch 17/100
12/12 [=====] - 118s 10s/step - loss: 1.1731 - acc:
0.5578 - precision: 0.6641 - recall: 0.3377 - val_loss: 1.0124 - val_acc: 0.6
328 - val_precision: 0.7964 - val_recall: 0.4193
Epoch 18/100
12/12 [=====] - 116s 10s/step - loss: 1.1139 - acc:
0.5811 - precision: 0.7152 - recall: 0.3822 - val_loss: 1.0054 - val_acc: 0.6
469 - val_precision: 0.7768 - val_recall: 0.4341
Epoch 19/100
12/12 [=====] - 115s 10s/step - loss: 1.2173 - acc:
0.5236 - precision: 0.7069 - recall: 0.3508 - val_loss: 1.0295 - val_acc: 0.6
328 - val_precision: 0.8714 - val_recall: 0.4036
Epoch 20/100
12/12 [=====] - 121s 10s/step - loss: 1.1701 - acc:
0.5469 - precision: 0.6569 - recall: 0.3229 - val_loss: 0.9817 - val_acc: 0.6
752 - val_precision: 0.8354 - val_recall: 0.4047
Epoch 21/100
12/12 [=====] - 117s 10s/step - loss: 1.1689 - acc:
0.5603 - precision: 0.7302 - recall: 0.3508 - val_loss: 1.0244 - val_acc: 0.6
255 - val_precision: 0.8350 - val_recall: 0.3753
Epoch 22/100
12/12 [=====] - 115s 10s/step - loss: 1.1763 - acc:
0.5499 - precision: 0.6932 - recall: 0.3587 - val_loss: 0.9530 - val_acc: 0.6
836 - val_precision: 0.8292 - val_recall: 0.4333
Epoch 23/100
12/12 [=====] - 119s 10s/step - loss: 1.0847 - acc:
0.5843 - precision: 0.7179 - recall: 0.4000 - val_loss: 0.9587 - val_acc: 0.6
544 - val_precision: 0.7621 - val_recall: 0.4047
Epoch 24/100
12/12 [=====] - 114s 10s/step - loss: 1.1352 - acc:
0.5630 - precision: 0.7303 - recall: 0.3534 - val_loss: 0.9710 - val_acc: 0.6
477 - val_precision: 0.8093 - val_recall: 0.5008
Epoch 25/100
12/12 [=====] - 116s 10s/step - loss: 1.0359 - acc:
0.5938 - precision: 0.7167 - recall: 0.4661 - val_loss: 0.8959 - val_acc: 0.6
984 - val_precision: 0.7772 - val_recall: 0.5146
Epoch 26/100
12/12 [=====] - 119s 10s/step - loss: 1.0052 - acc:
0.5862 - precision: 0.7550 - recall: 0.4554 - val_loss: 0.9121 - val_acc: 0.6
755 - val_precision: 0.8142 - val_recall: 0.5143
Epoch 27/100
12/12 [=====] - 114s 10s/step - loss: 1.0554 - acc:
0.5652 - precision: 0.6855 - recall: 0.3978 - val_loss: 0.9950 - val_acc: 0.6
466 - val_precision: 0.8295 - val_recall: 0.4557
Epoch 28/100
12/12 [=====] - 117s 10s/step - loss: 1.0493 - acc:
0.5916 - precision: 0.7127 - recall: 0.4111 - val_loss: 0.8831 - val_acc: 0.6
687 - val_precision: 0.8467 - val_recall: 0.5294
Epoch 29/100
12/12 [=====] - 118s 10s/step - loss: 0.9878 - acc:
```

0.5916 - precision: 0.7015 - recall: 0.4319 - val\_loss: 0.8515 - val\_acc: 0.7  
287 - val\_precision: 0.8121 - val\_recall: 0.5062  
Epoch 30/100  
12/12 [=====] - 115s 10s/step - loss: 0.9852 - acc:  
0.6125 - precision: 0.7120 - recall: 0.4504 - val\_loss: 0.8492 - val\_acc: 0.6  
906 - val\_precision: 0.7726 - val\_recall: 0.4995  
Epoch 31/100  
12/12 [=====] - 120s 10s/step - loss: 0.9026 - acc:  
0.6415 - precision: 0.7588 - recall: 0.4923 - val\_loss: 0.8770 - val\_acc: 0.6  
968 - val\_precision: 0.7808 - val\_recall: 0.5575  
Epoch 32/100  
12/12 [=====] - 117s 10s/step - loss: 0.9788 - acc:  
0.6153 - precision: 0.7054 - recall: 0.4897 - val\_loss: 0.9581 - val\_acc: 0.6  
461 - val\_precision: 0.7321 - val\_recall: 0.5429  
Epoch 33/100  
12/12 [=====] - 115s 10s/step - loss: 0.9493 - acc:  
0.6302 - precision: 0.7262 - recall: 0.4974 - val\_loss: 0.9772 - val\_acc: 0.6  
331 - val\_precision: 0.6873 - val\_recall: 0.5157  
Epoch 34/100  
12/12 [=====] - 119s 10s/step - loss: 1.0184 - acc:  
0.5574 - precision: 0.7132 - recall: 0.4371 - val\_loss: 0.9067 - val\_acc: 0.6  
690 - val\_precision: 0.8411 - val\_recall: 0.5070  
Epoch 35/100  
12/12 [=====] - 114s 10s/step - loss: 1.1074 - acc:  
0.5706 - precision: 0.6786 - recall: 0.4260 - val\_loss: 0.8651 - val\_acc: 0.7  
060 - val\_precision: 0.8649 - val\_recall: 0.5140  
Epoch 36/100  
12/12 [=====] - 117s 10s/step - loss: 0.9452 - acc:  
0.6224 - precision: 0.7466 - recall: 0.4323 - val\_loss: 0.7941 - val\_acc: 0.6  
909 - val\_precision: 0.7650 - val\_recall: 0.5516  
Epoch 37/100  
12/12 [=====] - 119s 10s/step - loss: 0.9497 - acc:  
0.6414 - precision: 0.7297 - recall: 0.5183 - val\_loss: 0.9096 - val\_acc: 0.6  
404 - val\_precision: 0.6801 - val\_recall: 0.5451  
Epoch 38/100  
12/12 [=====] - 115s 10s/step - loss: 0.9826 - acc:  
0.5892 - precision: 0.7309 - recall: 0.4872 - val\_loss: 0.7658 - val\_acc: 0.7  
716 - val\_precision: 0.8332 - val\_recall: 0.5888  
Epoch 39/100  
12/12 [=====] - 115s 10s/step - loss: 1.0370 - acc:  
0.6145 - precision: 0.6833 - recall: 0.4968 - val\_loss: 0.8335 - val\_acc: 0.7  
065 - val\_precision: 0.8525 - val\_recall: 0.5162  
Epoch 40/100  
12/12 [=====] - 120s 10s/step - loss: 0.9923 - acc:  
0.6230 - precision: 0.8019 - recall: 0.3794 - val\_loss: 0.7767 - val\_acc: 0.7  
427 - val\_precision: 0.8309 - val\_recall: 0.6099  
Epoch 41/100  
12/12 [=====] - 116s 10s/step - loss: 0.9211 - acc:  
0.6250 - precision: 0.7030 - recall: 0.4792 - val\_loss: 0.8000 - val\_acc: 0.7  
133 - val\_precision: 0.7978 - val\_recall: 0.5802  
Epoch 42/100  
12/12 [=====] - 116s 10s/step - loss: 0.9524 - acc:  
0.6423 - precision: 0.7529 - recall: 0.5105 - val\_loss: 0.7884 - val\_acc: 0.6  
463 - val\_precision: 0.7674 - val\_recall: 0.5362  
Epoch 43/100  
12/12 [=====] - 119s 10s/step - loss: 0.8740 - acc:  
0.6615 - precision: 0.7834 - recall: 0.5234 - val\_loss: 0.8791 - val\_acc: 0.6

```
471 - val_precision: 0.7595 - val_recall: 0.6029
Epoch 44/100
12/12 [=====] - 116s 10s/step - loss: 0.8882 - acc:
0.6659 - precision: 0.7665 - recall: 0.5420 - val_loss: 0.8982 - val_acc: 0.6
704 - val_precision: 0.7659 - val_recall: 0.5737
Epoch 45/100
12/12 [=====] - 118s 10s/step - loss: 0.9692 - acc:
0.6098 - precision: 0.6999 - recall: 0.4788 - val_loss: 0.8748 - val_acc: 0.6
687 - val_precision: 0.7761 - val_recall: 0.5880
Epoch 46/100
12/12 [=====] - 118s 10s/step - loss: 0.8574 - acc:
0.6693 - precision: 0.7681 - recall: 0.5130 - val_loss: 0.8164 - val_acc: 0.6
836 - val_precision: 0.7289 - val_recall: 0.5729
Epoch 47/100
12/12 [=====] - 115s 10s/step - loss: 0.8859 - acc:
0.6624 - precision: 0.7493 - recall: 0.5394 - val_loss: 0.9102 - val_acc: 0.6
544 - val_precision: 0.7038 - val_recall: 0.6099
Epoch 48/100
12/12 [=====] - 121s 10s/step - loss: 0.8652 - acc:
0.6750 - precision: 0.7683 - recall: 0.5965 - val_loss: 0.7732 - val_acc: 0.6
839 - val_precision: 0.7677 - val_recall: 0.5594
Epoch 49/100
12/12 [=====] - 115s 10s/step - loss: 0.8901 - acc:
0.6624 - precision: 0.7617 - recall: 0.5052 - val_loss: 0.7652 - val_acc: 0.7
206 - val_precision: 0.7676 - val_recall: 0.5734
Epoch 50/100
12/12 [=====] - 113s 9s/step - loss: 0.7658 - acc:
0.6888 - precision: 0.7681 - recall: 0.5840 - val_loss: 0.8939 - val_acc: 0.6
760 - val_precision: 0.6921 - val_recall: 0.5953
Epoch 51/100
12/12 [=====] - 118s 10s/step - loss: 0.9000 - acc:
0.6600 - precision: 0.7100 - recall: 0.5763 - val_loss: 0.8631 - val_acc: 0.6
552 - val_precision: 0.7207 - val_recall: 0.5513
Epoch 52/100
12/12 [=====] - 116s 10s/step - loss: 0.8229 - acc:
0.6675 - precision: 0.7670 - recall: 0.5314 - val_loss: 0.7432 - val_acc: 0.6
909 - val_precision: 0.7830 - val_recall: 0.5799
Epoch 53/100
12/12 [=====] - 115s 10s/step - loss: 0.7400 - acc:
0.6857 - precision: 0.7882 - recall: 0.5969 - val_loss: 0.7238 - val_acc: 0.6
917 - val_precision: 0.7717 - val_recall: 0.6185
Epoch 54/100
12/12 [=====] - 119s 10s/step - loss: 0.8906 - acc:
0.6599 - precision: 0.7479 - recall: 0.5500 - val_loss: 0.7667 - val_acc: 0.7
133 - val_precision: 0.8104 - val_recall: 0.6253
Epoch 55/100
12/12 [=====] - 115s 10s/step - loss: 0.8246 - acc:
0.6653 - precision: 0.7796 - recall: 0.5633 - val_loss: 0.7842 - val_acc: 0.6
782 - val_precision: 0.7806 - val_recall: 0.5967
Epoch 56/100
12/12 [=====] - 118s 10s/step - loss: 0.7868 - acc:
0.7161 - precision: 0.7987 - recall: 0.5885 - val_loss: 0.7803 - val_acc: 0.6
844 - val_precision: 0.7319 - val_recall: 0.5591
Epoch 57/100
12/12 [=====] - 116s 10s/step - loss: 0.8744 - acc:
0.6553 - precision: 0.7161 - recall: 0.5234 - val_loss: 0.8090 - val_acc: 0.6
617 - val_precision: 0.7620 - val_recall: 0.6102
```



Epoch 58/100  
12/12 [=====] - 114s 10s/step - loss: 0.7404 - acc: 0.6858 - precision: 0.7939 - recall: 0.6048 - val\_loss: 0.7975 - val\_acc: 0.6914 - val\_precision: 0.7570 - val\_recall: 0.5961

Epoch 59/100  
12/12 [=====] - 116s 10s/step - loss: 0.8434 - acc: 0.6807 - precision: 0.7779 - recall: 0.5550 - val\_loss: 0.7220 - val\_acc: 0.7352 - val\_precision: 0.7833 - val\_recall: 0.6172

Epoch 60/100  
12/12 [=====] - 117s 10s/step - loss: 0.7467 - acc: 0.7135 - precision: 0.8042 - recall: 0.6042 - val\_loss: 0.7156 - val\_acc: 0.7198 - val\_precision: 0.7804 - val\_recall: 0.6317

Epoch 61/100  
12/12 [=====] - 114s 10s/step - loss: 0.7391 - acc: 0.7117 - precision: 0.7572 - recall: 0.6202 - val\_loss: 0.7435 - val\_acc: 0.6911 - val\_precision: 0.7302 - val\_recall: 0.6169

Epoch 62/100  
12/12 [=====] - 120s 10s/step - loss: 0.8806 - acc: 0.6599 - precision: 0.7410 - recall: 0.5814 - val\_loss: 0.7583 - val\_acc: 0.6903 - val\_precision: 0.8010 - val\_recall: 0.6247

Epoch 63/100  
12/12 [=====] - 115s 10s/step - loss: 0.8606 - acc: 0.6704 - precision: 0.7636 - recall: 0.5181 - val\_loss: 0.8520 - val\_acc: 0.6258 - val\_precision: 0.7224 - val\_recall: 0.5518

Epoch 64/100  
12/12 [=====] - 114s 10s/step - loss: 0.7876 - acc: 0.6987 - precision: 0.7667 - recall: 0.5915 - val\_loss: 0.7682 - val\_acc: 0.7125 - val\_precision: 0.7964 - val\_recall: 0.6390

Epoch 65/100  
12/12 [=====] - 120s 10s/step - loss: 0.7480 - acc: 0.7383 - precision: 0.8189 - recall: 0.6102 - val\_loss: 0.8504 - val\_acc: 0.6539 - val\_precision: 0.7346 - val\_recall: 0.6099

Epoch 66/100  
12/12 [=====] - 115s 10s/step - loss: 0.7896 - acc: 0.7042 - precision: 0.7959 - recall: 0.6021 - val\_loss: 0.7771 - val\_acc: 0.6833 - val\_precision: 0.7532 - val\_recall: 0.6320

Epoch 67/100  
12/12 [=====] - 116s 10s/step - loss: 0.7584 - acc: 0.7095 - precision: 0.7922 - recall: 0.6179 - val\_loss: 0.7745 - val\_acc: 0.6768 - val\_precision: 0.7414 - val\_recall: 0.6331

Epoch 68/100  
12/12 [=====] - 118s 10s/step - loss: 0.7044 - acc: 0.6936 - precision: 0.7681 - recall: 0.6465 - val\_loss: 0.7943 - val\_acc: 0.6839 - val\_precision: 0.7196 - val\_recall: 0.6396

Epoch 69/100  
12/12 [=====] - 113s 9s/step - loss: 0.6945 - acc: 0.7173 - precision: 0.7690 - recall: 0.6361 - val\_loss: 0.6931 - val\_acc: 0.7276 - val\_precision: 0.8130 - val\_recall: 0.6388

Epoch 70/100  
12/12 [=====] - 116s 10s/step - loss: 0.7796 - acc: 0.7173 - precision: 0.7817 - recall: 0.6388 - val\_loss: 0.6690 - val\_acc: 0.7133 - val\_precision: 0.7933 - val\_recall: 0.6469

Epoch 71/100  
12/12 [=====] - 120s 10s/step - loss: 0.8320 - acc: 0.6615 - precision: 0.7460 - recall: 0.5547 - val\_loss: 0.7139 - val\_acc: 0.6898 - val\_precision: 0.7424 - val\_recall: 0.6166

Epoch 72/100

```
12/12 [=====] - 113s 9s/step - loss: 0.7029 - acc:
0.7291 - precision: 0.8337 - recall: 0.6370 - val_loss: 0.7370 - val_acc: 0.7
349 - val_precision: 0.8034 - val_recall: 0.6609
Epoch 73/100
12/12 [=====] - 119s 10s/step - loss: 0.6980 - acc:
0.7109 - precision: 0.7958 - recall: 0.6016 - val_loss: 0.7065 - val_acc: 0.7
198 - val_precision: 0.7776 - val_recall: 0.6976
Epoch 74/100
12/12 [=====] - 115s 10s/step - loss: 0.6770 - acc:
0.7133 - precision: 0.7900 - recall: 0.6554 - val_loss: 0.6677 - val_acc: 0.7
424 - val_precision: 0.7928 - val_recall: 0.6466
Epoch 75/100
12/12 [=====] - 116s 10s/step - loss: 0.6858 - acc:
0.7396 - precision: 0.8118 - recall: 0.6536 - val_loss: 0.6766 - val_acc: 0.7
424 - val_precision: 0.8014 - val_recall: 0.6833
Epoch 76/100
12/12 [=====] - 119s 10s/step - loss: 0.7352 - acc:
0.7304 - precision: 0.8009 - recall: 0.6204 - val_loss: 0.9542 - val_acc: 0.6
396 - val_precision: 0.6973 - val_recall: 0.5583
Epoch 77/100
12/12 [=====] - 115s 10s/step - loss: 0.7238 - acc:
0.7155 - precision: 0.7863 - recall: 0.6235 - val_loss: 0.7092 - val_acc: 0.7
060 - val_precision: 0.7622 - val_recall: 0.6323
Epoch 78/100
12/12 [=====] - 116s 10s/step - loss: 0.7432 - acc:
0.7188 - precision: 0.8033 - recall: 0.6484 - val_loss: 0.7962 - val_acc: 0.6
617 - val_precision: 0.7308 - val_recall: 0.6180
Epoch 79/100
12/12 [=====] - 119s 10s/step - loss: 0.7481 - acc:
0.7515 - precision: 0.8072 - recall: 0.6312 - val_loss: 0.7554 - val_acc: 0.6
830 - val_precision: 0.7713 - val_recall: 0.6466
Epoch 80/100
12/12 [=====] - 113s 9s/step - loss: 0.7523 - acc:
0.7307 - precision: 0.8177 - recall: 0.6074 - val_loss: 0.6978 - val_acc: 0.7
068 - val_precision: 0.7420 - val_recall: 0.6112
Epoch 81/100
12/12 [=====] - 115s 10s/step - loss: 0.6502 - acc:
0.7539 - precision: 0.8464 - recall: 0.6728 - val_loss: 0.7365 - val_acc: 0.7
130 - val_precision: 0.7584 - val_recall: 0.6469
Epoch 82/100
12/12 [=====] - 119s 10s/step - loss: 0.7649 - acc:
0.6831 - precision: 0.7505 - recall: 0.6358 - val_loss: 0.7251 - val_acc: 0.7
130 - val_precision: 0.7802 - val_recall: 0.6763
Epoch 83/100
12/12 [=====] - 114s 10s/step - loss: 0.7219 - acc:
0.7092 - precision: 0.7720 - recall: 0.6387 - val_loss: 0.6709 - val_acc: 0.7
138 - val_precision: 0.7461 - val_recall: 0.6477
Epoch 84/100
12/12 [=====] - 116s 10s/step - loss: 0.7020 - acc:
0.7408 - precision: 0.7907 - recall: 0.6779 - val_loss: 0.7237 - val_acc: 0.7
279 - val_precision: 0.7943 - val_recall: 0.6539
Epoch 85/100
12/12 [=====] - 117s 10s/step - loss: 0.7814 - acc:
0.6937 - precision: 0.7763 - recall: 0.5995 - val_loss: 0.7583 - val_acc: 0.6
911 - val_precision: 0.7927 - val_recall: 0.6466
Epoch 86/100
12/12 [=====] - 115s 10s/step - loss: 0.7498 - acc:
```

0.6729 - precision: 0.7576 - recall: 0.5707 - val\_loss: 0.6645 - val\_acc: 0.7  
497 - val\_precision: 0.8066 - val\_recall: 0.6763  
Epoch 87/100  
12/12 [=====] - 118s 10s/step - loss: 0.6082 - acc:  
0.7708 - precision: 0.8275 - recall: 0.6823 - val\_loss: 0.7959 - val\_acc: 0.6  
695 - val\_precision: 0.7111 - val\_recall: 0.6328  
Epoch 88/100  
12/12 [=====] - 114s 10s/step - loss: 0.6544 - acc:  
0.7686 - precision: 0.8142 - recall: 0.6897 - val\_loss: 0.6899 - val\_acc: 0.7  
570 - val\_precision: 0.7729 - val\_recall: 0.6544  
Epoch 89/100  
12/12 [=====] - 115s 10s/step - loss: 0.7180 - acc:  
0.7045 - precision: 0.7639 - recall: 0.6287 - val\_loss: 0.7851 - val\_acc: 0.6  
766 - val\_precision: 0.7158 - val\_recall: 0.6471  
Epoch 90/100  
12/12 [=====] - 119s 10s/step - loss: 0.7990 - acc:  
0.7188 - precision: 0.7834 - recall: 0.5990 - val\_loss: 0.7767 - val\_acc: 0.6  
901 - val\_precision: 0.7628 - val\_recall: 0.6382  
Epoch 91/100  
12/12 [=====] - 117s 10s/step - loss: 0.6907 - acc:  
0.7408 - precision: 0.8415 - recall: 0.6363 - val\_loss: 0.6711 - val\_acc: 0.7  
149 - val\_precision: 0.7525 - val\_recall: 0.6417  
Epoch 92/100  
12/12 [=====] - 113s 9s/step - loss: 0.6841 - acc:  
0.7489 - precision: 0.8118 - recall: 0.6858 - val\_loss: 0.7462 - val\_acc: 0.7  
130 - val\_precision: 0.7581 - val\_recall: 0.6690  
Epoch 93/100  
12/12 [=====] - 119s 10s/step - loss: 0.6870 - acc:  
0.7422 - precision: 0.7958 - recall: 0.6797 - val\_loss: 0.8159 - val\_acc: 0.6  
987 - val\_precision: 0.7220 - val\_recall: 0.6474  
Epoch 94/100  
12/12 [=====] - 114s 10s/step - loss: 0.6456 - acc:  
0.7616 - precision: 0.8315 - recall: 0.6570 - val\_loss: 0.6653 - val\_acc: 0.7  
427 - val\_precision: 0.7769 - val\_recall: 0.6917  
Epoch 95/100  
12/12 [=====] - 114s 10s/step - loss: 0.6358 - acc:  
0.7789 - precision: 0.8407 - recall: 0.7050 - val\_loss: 0.7152 - val\_acc: 0.7  
360 - val\_precision: 0.7844 - val\_recall: 0.7211  
Epoch 96/100  
12/12 [=====] - 119s 10s/step - loss: 0.5405 - acc:  
0.7839 - precision: 0.8386 - recall: 0.7344 - val\_loss: 0.7804 - val\_acc: 0.7  
141 - val\_precision: 0.7717 - val\_recall: 0.6920  
Epoch 97/100  
12/12 [=====] - 113s 9s/step - loss: 0.5856 - acc:  
0.7687 - precision: 0.8242 - recall: 0.7109 - val\_loss: 0.6749 - val\_acc: 0.7  
495 - val\_precision: 0.7854 - val\_recall: 0.6979  
Epoch 98/100  
12/12 [=====] - 118s 10s/step - loss: 0.6013 - acc:  
0.7671 - precision: 0.8251 - recall: 0.7122 - val\_loss: 0.7112 - val\_acc: 0.7  
357 - val\_precision: 0.7542 - val\_recall: 0.6987  
Epoch 99/100  
12/12 [=====] - 117s 10s/step - loss: 0.6464 - acc:  
0.7318 - precision: 0.7987 - recall: 0.6719 - val\_loss: 0.7496 - val\_acc: 0.7  
208 - val\_precision: 0.7591 - val\_recall: 0.6914  
Epoch 100/100  
12/12 [=====] - 113s 9s/step - loss: 0.6777 - acc:

0.7451 - precision: 0.8063 - recall: 0.6557 - val\_loss: 0.7176 - val\_acc: 0.7063 - val\_precision: 0.7553 - val\_recall: 0.6552

```
In [5]: test_steps_per_epoch = np.math.ceil(test_set.samples / test_set.batch_size)
        predictions = classifier.predict_generator(test_set, steps=test_steps_per_epoch)
        predicted_classes = np.argmax(predictions, axis=1)
```

```
In [6]: true_classes = test_set.classes
        class_labels = list(test_set.class_indices.keys())
```

```
In [7]: import sklearn.metrics as metrics
        report = metrics.classification_report(true_classes, predicted_classes, target_names=class_labels)
        print(report)
```

	precision	recall	f1-score	support
anger	0.24	0.25	0.25	32
boredom	0.10	0.05	0.06	21
disgust	0.11	0.17	0.13	12
fear	0.20	0.12	0.15	17
happiness	0.16	0.17	0.16	18
neutral	0.13	0.20	0.16	20
sadness	0.21	0.19	0.20	16
avg / total	0.17	0.17	0.17	136

```

In [10]: import matplotlib.pyplot as plt
import itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting normalize=True.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]*100
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap, aspect = 'auto')
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

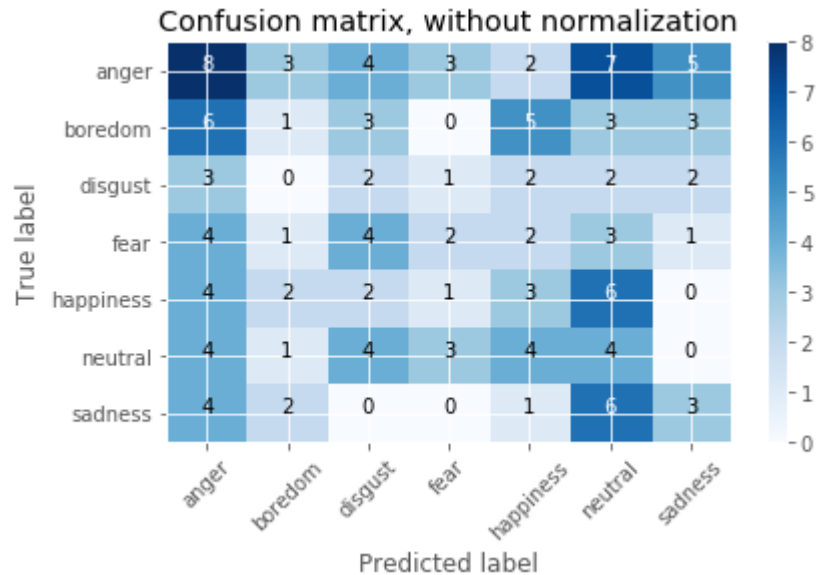
# Compute confusion matrix
cnf_matrix = metrics.confusion_matrix(true_classes, predicted_classes)
np.set_printoptions(precision=4)

# Plot non-normalized confusion matrix
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_labels,
                      title='Confusion matrix, without normalization')
plt.savefig("non_normalized_confusion_matrix_cnn.png")
plt.show()
# Plot normalized confusion matrix
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_labels, normalize=True,
                      title='Normalized confusion matrix')
plt.savefig("normalized_confusion_matrix_cnn.png")
plt.show()

```

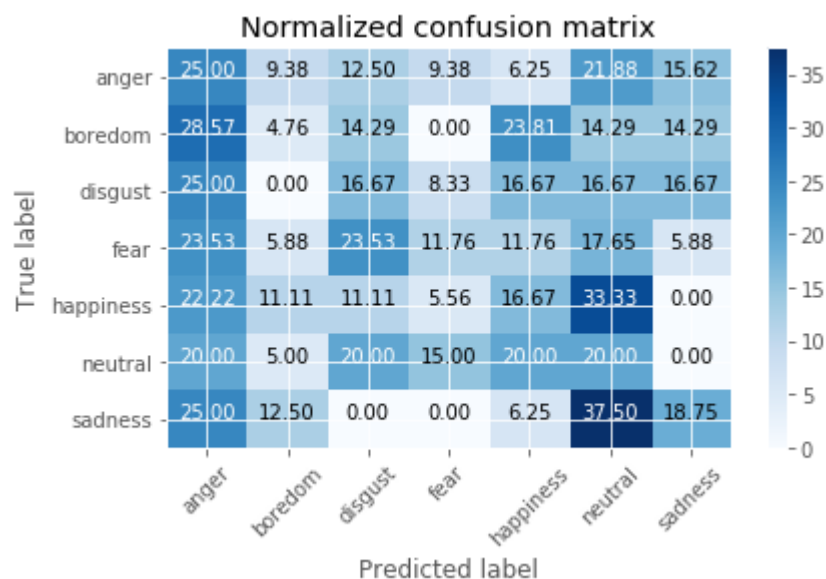
Confusion matrix, without normalization

```
[[8 3 4 3 2 7 5]
 [6 1 3 0 5 3 3]
 [3 0 2 1 2 2 2]
 [4 1 4 2 2 3 1]
 [4 2 2 1 3 6 0]
 [4 1 4 3 4 4 0]
 [4 2 0 0 1 6 3]]
```



Normalized confusion matrix

```
[[25.      9.375 12.5     9.375  6.25  21.875 15.625 ]
 [28.5714  4.7619 14.2857  0.      23.8095 14.2857 14.2857]
 [25.       0.     16.6667  8.3333 16.6667 16.6667 16.6667]
 [23.5294  5.8824 23.5294 11.7647 11.7647 17.6471  5.8824]
 [22.2222 11.1111 11.1111  5.5556 16.6667 33.3333  0.      ]
 [20.       5.     20.      15.     20.     20.     0.      ]
 [25.      12.5    0.       0.       6.25   37.5    18.75   ]]
```



```
In [11]: import matplotlib.pyplot as plt
plt.style.use("ggplot")
plt.figure()
N = 100
plt.plot(np.arange(0, N), results.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), results.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), results.history["acc"], label="train_acc")
plt.plot(np.arange(0, N), results.history["val_acc"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="upper left")
plt.savefig("plot_cnn.png")
```

