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 conolutional 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(Reshape((4*4, 1024)))
classifier.add(LSTM(units = 50, return_sequences = True, dropout = 0.5))
classifier.add(LSTM(units = 20, return_sequences = False, dropout = 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 depreca
ted. 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
reshape_1 (Reshape)	(None, 16, 1024)	0
lstm_1 (LSTM)	(None, 16, 50)	215000
lstm_2 (LSTM)	(None, 20)	5680
dense_1 (Dense)	(None, 7)	147

```
Total params: 296,475
Trainable params: 296,475
Non-trainable params: 0
```

In [2]:
```
# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metr
ics = ['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 4410 images belonging to 7 classes.
Found 1475 images belonging to 7 classes.
```

In [4]:
```
results = classifier.fit_generator(training_set,
                         samples_per_epoch = 4410,
                         nb_epoch = 100,
                         validation_data = test_set,
                         nb_val_samples = 1475)
```

```
Epoch 1/100
137/137 [==============================] - 951s 7s/step - loss: 1.8649 - acc:
0.2567 - precision: 0.1180 - recall: 0.0089 - val_loss: 1.6272 - val_acc: 0.3
793 - val_precision: 0.6935 - val_recall: 0.0694
Epoch 2/100
137/137 [==============================] - 1188s 9s/step - loss: 1.5503 - ac
c: 0.3998 - precision: 0.6108 - recall: 0.1104 - val_loss: 1.2741 - val_acc:
0.5081 - val_precision: 0.7049 - val_recall: 0.2695
Epoch 3/100
137/137 [==============================] - 1874s 14s/step - loss: 1.3563 - ac
c: 0.4855 - precision: 0.6746 - recall: 0.2355 - val_loss: 1.2293 - val_acc:
0.5370 - val_precision: 0.6147 - val_recall: 0.4085
Epoch 4/100
137/137 [==============================] - 1872s 14s/step - loss: 1.2445 - ac
c: 0.5266 - precision: 0.6832 - recall: 0.3071 - val_loss: 1.0961 - val_acc:
0.5927 - val_precision: 0.6834 - val_recall: 0.4644
Epoch 5/100
137/137 [==============================] - 1876s 14s/step - loss: 1.1746 - ac
c: 0.5598 - precision: 0.7032 - recall: 0.3556 - val_loss: 0.9661 - val_acc:
0.6319 - val_precision: 0.7463 - val_recall: 0.5039
Epoch 6/100
137/137 [==============================] - 1873s 14s/step - loss: 1.0817 - ac
c: 0.5916 - precision: 0.7355 - recall: 0.4055 - val_loss: 0.8652 - val_acc:
0.6754 - val_precision: 0.7580 - val_recall: 0.5588
Epoch 7/100
137/137 [==============================] - 1876s 14s/step - loss: 1.0435 - ac
c: 0.6006 - precision: 0.7187 - recall: 0.4353 - val_loss: 0.8128 - val_acc:
0.6947 - val_precision: 0.7855 - val_recall: 0.5862
Epoch 8/100
137/137 [==============================] - 1886s 14s/step - loss: 0.9913 - ac
c: 0.6213 - precision: 0.7309 - recall: 0.4800 - val_loss: 0.7993 - val_acc:
0.7014 - val_precision: 0.7722 - val_recall: 0.6111
Epoch 9/100
137/137 [==============================] - 1891s 14s/step - loss: 0.9482 - ac
c: 0.6434 - precision: 0.7391 - recall: 0.5171 - val_loss: 0.7361 - val_acc:
0.7182 - val_precision: 0.7827 - val_recall: 0.6395
Epoch 10/100
137/137 [==============================] - 1879s 14s/step - loss: 0.9000 - ac
c: 0.6603 - precision: 0.7461 - recall: 0.5501 - val_loss: 0.6941 - val_acc:
0.7532 - val_precision: 0.8119 - val_recall: 0.6874
Epoch 11/100
137/137 [==============================] - 1872s 14s/step - loss: 0.8997 - ac
c: 0.6551 - precision: 0.7355 - recall: 0.5538 - val_loss: 0.7008 - val_acc:
0.7292 - val_precision: 0.7838 - val_recall: 0.6662
Epoch 12/100
137/137 [==============================] - 1875s 14s/step - loss: 0.8642 - ac
c: 0.6794 - precision: 0.7506 - recall: 0.5680 - val_loss: 0.6830 - val_acc:
0.7488 - val_precision: 0.7861 - val_recall: 0.6948
Epoch 13/100
137/137 [==============================] - 1925s 14s/step - loss: 0.8407 - ac
c: 0.6806 - precision: 0.7447 - recall: 0.5918 - val_loss: 0.6254 - val_acc:
0.7764 - val_precision: 0.8196 - val_recall: 0.7275
Epoch 14/100
137/137 [==============================] - 1878s 14s/step - loss: 0.7986 - ac
c: 0.6913 - precision: 0.7602 - recall: 0.6094 - val_loss: 0.5857 - val_acc:
0.7732 - val_precision: 0.8297 - val_recall: 0.7338
Epoch 15/100
```

```
137/137 [==============================] - 1888s 14s/step - loss: 0.7804 - ac
c: 0.7063 - precision: 0.7761 - recall: 0.6249 - val_loss: 0.6664 - val_acc:
0.7519 - val_precision: 0.7890 - val_recall: 0.7045
Epoch 16/100
137/137 [==============================] - 1882s 14s/step - loss: 0.7566 - ac
c: 0.7142 - precision: 0.7774 - recall: 0.6363 - val_loss: 0.5372 - val_acc:
0.8081 - val_precision: 0.8442 - val_recall: 0.7648
Epoch 17/100
137/137 [==============================] - 1887s 14s/step - loss: 0.7385 - ac
c: 0.7222 - precision: 0.7770 - recall: 0.6503 - val_loss: 0.5412 - val_acc:
0.7969 - val_precision: 0.8322 - val_recall: 0.7572
Epoch 18/100
137/137 [==============================] - 1875s 14s/step - loss: 0.7238 - ac
c: 0.7276 - precision: 0.7799 - recall: 0.6613 - val_loss: 0.5011 - val_acc:
0.8110 - val_precision: 0.8465 - val_recall: 0.7730
Epoch 19/100
137/137 [==============================] - 1876s 14s/step - loss: 0.6722 - ac
c: 0.7476 - precision: 0.8073 - recall: 0.6849 - val_loss: 0.4288 - val_acc:
0.8447 - val_precision: 0.8805 - val_recall: 0.8136
Epoch 20/100
137/137 [==============================] - 1879s 14s/step - loss: 0.6827 - ac
c: 0.7487 - precision: 0.7942 - recall: 0.6825 - val_loss: 0.4378 - val_acc:
0.8389 - val_precision: 0.8731 - val_recall: 0.8097
Epoch 21/100
137/137 [==============================] - 1880s 14s/step - loss: 0.6364 - ac
c: 0.7637 - precision: 0.8092 - recall: 0.7100 - val_loss: 0.5182 - val_acc:
0.8040 - val_precision: 0.8339 - val_recall: 0.7688
Epoch 22/100
137/137 [==============================] - 1884s 14s/step - loss: 0.6469 - ac
c: 0.7642 - precision: 0.8051 - recall: 0.7075 - val_loss: 0.4086 - val_acc:
0.8522 - val_precision: 0.8726 - val_recall: 0.8271
Epoch 23/100
137/137 [==============================] - 1885s 14s/step - loss: 0.5875 - ac
c: 0.7796 - precision: 0.8201 - recall: 0.7298 - val_loss: 0.5004 - val_acc:
0.7928 - val_precision: 0.8271 - val_recall: 0.7670
Epoch 24/100
137/137 [==============================] - 1887s 14s/step - loss: 0.6027 - ac
c: 0.7781 - precision: 0.8129 - recall: 0.7338 - val_loss: 0.3664 - val_acc:
0.8714 - val_precision: 0.8929 - val_recall: 0.8464
Epoch 25/100
137/137 [==============================] - 1885s 14s/step - loss: 0.5762 - ac
c: 0.7861 - precision: 0.8247 - recall: 0.7456 - val_loss: 0.4049 - val_acc:
0.8592 - val_precision: 0.8778 - val_recall: 0.8361
Epoch 26/100
137/137 [==============================] - 1887s 14s/step - loss: 0.5606 - ac
c: 0.7937 - precision: 0.8230 - recall: 0.7476 - val_loss: 0.3281 - val_acc:
0.8854 - val_precision: 0.9004 - val_recall: 0.8650
Epoch 27/100
137/137 [==============================] - 1916s 14s/step - loss: 0.5535 - ac
c: 0.7911 - precision: 0.8306 - recall: 0.7546 - val_loss: 0.3162 - val_acc:
0.8930 - val_precision: 0.9070 - val_recall: 0.8746
Epoch 28/100
137/137 [==============================] - 1881s 14s/step - loss: 0.5332 - ac
c: 0.8005 - precision: 0.8324 - recall: 0.7635 - val_loss: 0.3138 - val_acc:
0.8960 - val_precision: 0.9117 - val_recall: 0.8777
Epoch 29/100
137/137 [==============================] - 1876s 14s/step - loss: 0.5233 - ac
```

```
c: 0.8109 - precision: 0.8400 - recall: 0.7724 - val_loss: 0.2823 - val_acc:
0.8983 - val_precision: 0.9166 - val_recall: 0.8868
Epoch 30/100
137/137 [==============================] - 1879s 14s/step - loss: 0.4979 - ac
c: 0.8187 - precision: 0.8403 - recall: 0.7922 - val_loss: 0.2683 - val_acc:
0.9010 - val_precision: 0.9133 - val_recall: 0.8928
Epoch 31/100
137/137 [==============================] - 1881s 14s/step - loss: 0.5096 - ac
c: 0.8121 - precision: 0.8402 - recall: 0.7792 - val_loss: 0.2355 - val_acc:
0.9205 - val_precision: 0.9306 - val_recall: 0.9110
Epoch 32/100
137/137 [==============================] - 1878s 14s/step - loss: 0.4844 - ac
c: 0.8190 - precision: 0.8448 - recall: 0.7906 - val_loss: 0.3058 - val_acc:
0.9010 - val_precision: 0.9146 - val_recall: 0.8867
Epoch 33/100
137/137 [==============================] - 1882s 14s/step - loss: 0.4535 - ac
c: 0.8360 - precision: 0.8625 - recall: 0.8104 - val_loss: 0.2354 - val_acc:
0.9227 - val_precision: 0.9295 - val_recall: 0.9125
Epoch 34/100
137/137 [==============================] - 1882s 14s/step - loss: 0.4631 - ac
c: 0.8285 - precision: 0.8572 - recall: 0.8022 - val_loss: 0.2215 - val_acc:
0.9309 - val_precision: 0.9398 - val_recall: 0.9194
Epoch 35/100
137/137 [==============================] - 1879s 14s/step - loss: 0.4452 - ac
c: 0.8397 - precision: 0.8600 - recall: 0.8176 - val_loss: 0.1715 - val_acc:
0.9505 - val_precision: 0.9546 - val_recall: 0.9416
Epoch 36/100
137/137 [==============================] - 1879s 14s/step - loss: 0.4330 - ac
c: 0.8454 - precision: 0.8664 - recall: 0.8235 - val_loss: 0.2505 - val_acc:
0.9078 - val_precision: 0.9235 - val_recall: 0.8990
Epoch 37/100
137/137 [==============================] - 1882s 14s/step - loss: 0.4326 - ac
c: 0.8410 - precision: 0.8610 - recall: 0.8180 - val_loss: 0.1650 - val_acc:
0.9458 - val_precision: 0.9513 - val_recall: 0.9410
Epoch 38/100
137/137 [==============================] - 1877s 14s/step - loss: 0.4306 - ac
c: 0.8392 - precision: 0.8600 - recall: 0.8182 - val_loss: 0.1679 - val_acc:
0.9450 - val_precision: 0.9510 - val_recall: 0.9362
Epoch 39/100
137/137 [==============================] - 1879s 14s/step - loss: 0.3983 - ac
c: 0.8553 - precision: 0.8745 - recall: 0.8375 - val_loss: 0.1820 - val_acc:
0.9404 - val_precision: 0.9470 - val_recall: 0.9316
Epoch 40/100
137/137 [==============================] - 1875s 14s/step - loss: 0.3821 - ac
c: 0.8623 - precision: 0.8797 - recall: 0.8441 - val_loss: 0.1985 - val_acc:
0.9296 - val_precision: 0.9368 - val_recall: 0.9241
Epoch 41/100
137/137 [==============================] - 1891s 14s/step - loss: 0.4169 - ac
c: 0.8497 - precision: 0.8711 - recall: 0.8280 - val_loss: 0.1589 - val_acc:
0.9485 - val_precision: 0.9534 - val_recall: 0.9417
Epoch 42/100
137/137 [==============================] - 1881s 14s/step - loss: 0.3842 - ac
c: 0.8657 - precision: 0.8847 - recall: 0.8467 - val_loss: 0.1883 - val_acc:
0.9471 - val_precision: 0.9501 - val_recall: 0.9431
Epoch 43/100
137/137 [==============================] - 1953s 14s/step - loss: 0.3444 - ac
c: 0.8755 - precision: 0.8910 - recall: 0.8591 - val_loss: 0.1625 - val_acc:
```

```
0.9532 - val_precision: 0.9568 - val_recall: 0.9470
Epoch 44/100
137/137 [==============================] - 1925s 14s/step - loss: 0.3656 - ac
c: 0.8705 - precision: 0.8864 - recall: 0.8566 - val_loss: 0.1656 - val_acc:
0.9404 - val_precision: 0.9449 - val_recall: 0.9404
Epoch 45/100
137/137 [==============================] - 1899s 14s/step - loss: 0.3642 - ac
c: 0.8695 - precision: 0.8846 - recall: 0.8533 - val_loss: 0.1417 - val_acc:
0.9477 - val_precision: 0.9528 - val_recall: 0.9457
Epoch 46/100
137/137 [==============================] - 1895s 14s/step - loss: 0.3379 - ac
c: 0.8769 - precision: 0.8909 - recall: 0.8621 - val_loss: 0.1063 - val_acc:
0.9702 - val_precision: 0.9720 - val_recall: 0.9674
Epoch 47/100
137/137 [==============================] - 1892s 14s/step - loss: 0.3693 - ac
c: 0.8667 - precision: 0.8819 - recall: 0.8523 - val_loss: 0.1298 - val_acc:
0.9565 - val_precision: 0.9622 - val_recall: 0.9524
Epoch 48/100
137/137 [==============================] - 1887s 14s/step - loss: 0.3289 - ac
c: 0.8853 - precision: 0.8996 - recall: 0.8718 - val_loss: 0.1213 - val_acc:
0.9607 - val_precision: 0.9639 - val_recall: 0.9587
Epoch 49/100
137/137 [==============================] - 1895s 14s/step - loss: 0.3176 - ac
c: 0.8893 - precision: 0.9017 - recall: 0.8783 - val_loss: 0.1058 - val_acc:
0.9689 - val_precision: 0.9734 - val_recall: 0.9668
Epoch 50/100
137/137 [==============================] - 1889s 14s/step - loss: 0.3151 - ac
c: 0.8871 - precision: 0.8996 - recall: 0.8732 - val_loss: 0.1193 - val_acc:
0.9649 - val_precision: 0.9687 - val_recall: 0.9629
Epoch 51/100
137/137 [==============================] - 1901s 14s/step - loss: 0.3334 - ac
c: 0.8797 - precision: 0.8907 - recall: 0.8670 - val_loss: 0.0967 - val_acc:
0.9722 - val_precision: 0.9735 - val_recall: 0.9708
Epoch 52/100
137/137 [==============================] - 1905s 14s/step - loss: 0.3232 - ac
c: 0.8889 - precision: 0.8985 - recall: 0.8752 - val_loss: 0.1006 - val_acc:
0.9717 - val_precision: 0.9729 - val_recall: 0.9690
Epoch 53/100
137/137 [==============================] - 1891s 14s/step - loss: 0.3003 - ac
c: 0.8942 - precision: 0.9061 - recall: 0.8795 - val_loss: 0.1246 - val_acc:
0.9572 - val_precision: 0.9604 - val_recall: 0.9559
Epoch 54/100
137/137 [==============================] - 1898s 14s/step - loss: 0.3073 - ac
c: 0.8941 - precision: 0.9064 - recall: 0.8809 - val_loss: 0.1031 - val_acc:
0.9662 - val_precision: 0.9667 - val_recall: 0.9641
Epoch 55/100
137/137 [==============================] - 1886s 14s/step - loss: 0.3102 - ac
c: 0.8879 - precision: 0.8988 - recall: 0.8768 - val_loss: 0.0941 - val_acc:
0.9749 - val_precision: 0.9768 - val_recall: 0.9722
Epoch 56/100
137/137 [==============================] - 1878s 14s/step - loss: 0.2910 - ac
c: 0.8948 - precision: 0.9036 - recall: 0.8848 - val_loss: 0.0875 - val_acc:
0.9703 - val_precision: 0.9708 - val_recall: 0.9682
Epoch 57/100
137/137 [==============================] - 1879s 14s/step - loss: 0.3043 - ac
c: 0.8930 - precision: 0.9034 - recall: 0.8841 - val_loss: 0.0870 - val_acc:
0.9728 - val_precision: 0.9754 - val_recall: 0.9715
```

```
Epoch 58/100
137/137 [==============================] - 1905s 14s/step - loss: 0.2573 - ac
c: 0.9101 - precision: 0.9203 - recall: 0.9017 - val_loss: 0.0937 - val_acc:
0.9756 - val_precision: 0.9775 - val_recall: 0.9722
Epoch 59/100
137/137 [==============================] - 1893s 14s/step - loss: 0.2723 - ac
c: 0.9061 - precision: 0.9166 - recall: 0.8942 - val_loss: 0.1056 - val_acc:
0.9694 - val_precision: 0.9727 - val_recall: 0.9688
Epoch 60/100
137/137 [==============================] - 1894s 14s/step - loss: 0.2714 - ac
c: 0.9056 - precision: 0.9152 - recall: 0.8946 - val_loss: 0.0593 - val_acc:
0.9851 - val_precision: 0.9857 - val_recall: 0.9830
Epoch 61/100
137/137 [==============================] - 1889s 14s/step - loss: 0.2563 - ac
c: 0.9096 - precision: 0.9199 - recall: 0.9012 - val_loss: 0.0647 - val_acc:
0.9791 - val_precision: 0.9837 - val_recall: 0.9770
Epoch 62/100
137/137 [==============================] - 1892s 14s/step - loss: 0.2633 - ac
c: 0.9005 - precision: 0.9110 - recall: 0.8946 - val_loss: 0.0772 - val_acc:
0.9790 - val_precision: 0.9803 - val_recall: 0.9776
Epoch 63/100
137/137 [==============================] - 1883s 14s/step - loss: 0.2497 - ac
c: 0.9159 - precision: 0.9253 - recall: 0.9102 - val_loss: 0.0843 - val_acc:
0.9749 - val_precision: 0.9762 - val_recall: 0.9735
Epoch 64/100
137/137 [==============================] - 1876s 14s/step - loss: 0.2728 - ac
c: 0.9005 - precision: 0.9117 - recall: 0.8918 - val_loss: 0.0731 - val_acc:
0.9775 - val_precision: 0.9789 - val_recall: 0.9769
Epoch 65/100
137/137 [==============================] - 1889s 14s/step - loss: 0.2537 - ac
c: 0.9105 - precision: 0.9205 - recall: 0.9018 - val_loss: 0.0501 - val_acc:
0.9858 - val_precision: 0.9878 - val_recall: 0.9838
Epoch 66/100
137/137 [==============================] - 1898s 14s/step - loss: 0.2407 - ac
c: 0.9149 - precision: 0.9227 - recall: 0.9078 - val_loss: 0.0493 - val_acc:
0.9851 - val_precision: 0.9864 - val_recall: 0.9838
Epoch 67/100
137/137 [==============================] - 1890s 14s/step - loss: 0.2603 - ac
c: 0.9030 - precision: 0.9109 - recall: 0.8971 - val_loss: 0.0987 - val_acc:
0.9695 - val_precision: 0.9728 - val_recall: 0.9675
Epoch 68/100
137/137 [==============================] - 1887s 14s/step - loss: 0.2604 - ac
c: 0.9016 - precision: 0.9133 - recall: 0.8932 - val_loss: 0.0994 - val_acc:
0.9668 - val_precision: 0.9700 - val_recall: 0.9647
Epoch 69/100
137/137 [==============================] - 1888s 14s/step - loss: 0.2303 - ac
c: 0.9204 - precision: 0.9274 - recall: 0.9113 - val_loss: 0.0480 - val_acc:
0.9898 - val_precision: 0.9918 - val_recall: 0.9857
Epoch 70/100
137/137 [==============================] - 1884s 14s/step - loss: 0.2280 - ac
c: 0.9227 - precision: 0.9297 - recall: 0.9142 - val_loss: 0.0749 - val_acc:
0.9797 - val_precision: 0.9804 - val_recall: 0.9797
Epoch 71/100
137/137 [==============================] - 1885s 14s/step - loss: 0.2570 - ac
c: 0.9089 - precision: 0.9182 - recall: 0.9022 - val_loss: 0.0379 - val_acc:
0.9939 - val_precision: 0.9939 - val_recall: 0.9925
Epoch 72/100
```

```
137/137 [==============================] - 1895s 14s/step - loss: 0.2131 - ac
c: 0.9245 - precision: 0.9326 - recall: 0.9177 - val_loss: 0.0503 - val_acc:
0.9898 - val_precision: 0.9905 - val_recall: 0.9885
Epoch 73/100
137/137 [==============================] - 1890s 14s/step - loss: 0.2099 - ac
c: 0.9259 - precision: 0.9330 - recall: 0.9189 - val_loss: 0.0808 - val_acc:
0.9742 - val_precision: 0.9742 - val_recall: 0.9742
Epoch 74/100
137/137 [==============================] - 1900s 14s/step - loss: 0.2221 - ac
c: 0.9240 - precision: 0.9304 - recall: 0.9199 - val_loss: 0.0734 - val_acc:
0.9756 - val_precision: 0.9769 - val_recall: 0.9756
Epoch 75/100
137/137 [==============================] - 1899s 14s/step - loss: 0.2003 - ac
c: 0.9306 - precision: 0.9377 - recall: 0.9251 - val_loss: 0.0625 - val_acc:
0.9802 - val_precision: 0.9802 - val_recall: 0.9802
Epoch 76/100
137/137 [==============================] - 1894s 14s/step - loss: 0.2274 - ac
c: 0.9200 - precision: 0.9267 - recall: 0.9173 - val_loss: 0.0534 - val_acc:
0.9824 - val_precision: 0.9843 - val_recall: 0.9810
Epoch 77/100
137/137 [==============================] - 1894s 14s/step - loss: 0.2042 - ac
c: 0.9308 - precision: 0.9380 - recall: 0.9228 - val_loss: 0.0481 - val_acc:
0.9884 - val_precision: 0.9891 - val_recall: 0.9871
Epoch 78/100
137/137 [==============================] - 1884s 14s/step - loss: 0.2125 - ac
c: 0.9274 - precision: 0.9341 - recall: 0.9217 - val_loss: 0.0628 - val_acc:
0.9778 - val_precision: 0.9784 - val_recall: 0.9771
Epoch 79/100
137/137 [==============================] - 1891s 14s/step - loss: 0.2169 - ac
c: 0.9243 - precision: 0.9293 - recall: 0.9186 - val_loss: 0.0604 - val_acc:
0.9811 - val_precision: 0.9810 - val_recall: 0.9791
Epoch 80/100
137/137 [==============================] - 1900s 14s/step - loss: 0.2087 - ac
c: 0.9286 - precision: 0.9355 - recall: 0.9243 - val_loss: 0.0483 - val_acc:
0.9851 - val_precision: 0.9851 - val_recall: 0.9844
Epoch 81/100
137/137 [==============================] - 1936s 14s/step - loss: 0.1815 - ac
c: 0.9386 - precision: 0.9441 - recall: 0.9339 - val_loss: 0.0406 - val_acc:
0.9864 - val_precision: 0.9878 - val_recall: 0.9864
Epoch 82/100
137/137 [==============================] - 1902s 14s/step - loss: 0.2099 - ac
c: 0.9287 - precision: 0.9324 - recall: 0.9214 - val_loss: 0.0741 - val_acc:
0.9728 - val_precision: 0.9734 - val_recall: 0.9721
Epoch 83/100
137/137 [==============================] - 1900s 14s/step - loss: 0.2027 - ac
c: 0.9323 - precision: 0.9396 - recall: 0.9263 - val_loss: 0.0678 - val_acc:
0.9844 - val_precision: 0.9844 - val_recall: 0.9837
Epoch 84/100
137/137 [==============================] - 1904s 14s/step - loss: 0.1862 - ac
c: 0.9353 - precision: 0.9412 - recall: 0.9287 - val_loss: 0.0401 - val_acc:
0.9891 - val_precision: 0.9905 - val_recall: 0.9891
Epoch 85/100
137/137 [==============================] - 1889s 14s/step - loss: 0.1931 - ac
c: 0.9355 - precision: 0.9407 - recall: 0.9291 - val_loss: 0.0602 - val_acc:
0.9824 - val_precision: 0.9837 - val_recall: 0.9817
Epoch 86/100
137/137 [==============================] - 1892s 14s/step - loss: 0.2099 - ac
```

```
c: 0.9303 - precision: 0.9374 - recall: 0.9243 - val_loss: 0.0597 - val_acc:
0.9783 - val_precision: 0.9782 - val_recall: 0.9769
Epoch 87/100
137/137 [==============================] - 1899s 14s/step - loss: 0.2076 - ac
c: 0.9290 - precision: 0.9335 - recall: 0.9237 - val_loss: 0.0375 - val_acc:
0.9899 - val_precision: 0.9899 - val_recall: 0.9899
Epoch 88/100
137/137 [==============================] - 1907s 14s/step - loss: 0.1671 - ac
c: 0.9401 - precision: 0.9453 - recall: 0.9344 - val_loss: 0.0318 - val_acc:
0.9932 - val_precision: 0.9932 - val_recall: 0.9932
Epoch 89/100
137/137 [==============================] - 1889s 14s/step - loss: 0.1786 - ac
c: 0.9391 - precision: 0.9442 - recall: 0.9332 - val_loss: 0.0461 - val_acc:
0.9898 - val_precision: 0.9898 - val_recall: 0.9885
Epoch 90/100
137/137 [==============================] - 1700s 12s/step - loss: 0.1970 - ac
c: 0.9376 - precision: 0.9428 - recall: 0.9330 - val_loss: 0.0340 - val_acc:
0.9932 - val_precision: 0.9939 - val_recall: 0.9925
Epoch 91/100
137/137 [==============================] - 1414s 10s/step - loss: 0.1675 - ac
c: 0.9408 - precision: 0.9464 - recall: 0.9360 - val_loss: 0.0769 - val_acc:
0.9714 - val_precision: 0.9721 - val_recall: 0.9708
Epoch 92/100
137/137 [==============================] - 987s 7s/step - loss: 0.1857 - acc:
0.9364 - precision: 0.9408 - recall: 0.9309 - val_loss: 0.0440 - val_acc: 0.9
885 - val_precision: 0.9885 - val_recall: 0.9871
Epoch 93/100
137/137 [==============================] - 984s 7s/step - loss: 0.1816 - acc:
0.9390 - precision: 0.9440 - recall: 0.9347 - val_loss: 0.0349 - val_acc: 0.9
898 - val_precision: 0.9905 - val_recall: 0.9885
Epoch 94/100
137/137 [==============================] - 985s 7s/step - loss: 0.1966 - acc:
0.9327 - precision: 0.9390 - recall: 0.9284 - val_loss: 0.1242 - val_acc: 0.9
627 - val_precision: 0.9632 - val_recall: 0.9593
Epoch 95/100
137/137 [==============================] - 987s 7s/step - loss: 0.1928 - acc:
0.9350 - precision: 0.9416 - recall: 0.9307 - val_loss: 0.0314 - val_acc: 0.9
932 - val_precision: 0.9939 - val_recall: 0.9932
Epoch 96/100
137/137 [==============================] - 988s 7s/step - loss: 0.1618 - acc:
0.9425 - precision: 0.9472 - recall: 0.9405 - val_loss: 0.0376 - val_acc: 0.9
919 - val_precision: 0.9919 - val_recall: 0.9919
Epoch 97/100
137/137 [==============================] - 987s 7s/step - loss: 0.1638 - acc:
0.9446 - precision: 0.9484 - recall: 0.9410 - val_loss: 0.0201 - val_acc: 0.9
946 - val_precision: 0.9946 - val_recall: 0.9946
Epoch 98/100
137/137 [==============================] - 986s 7s/step - loss: 0.1553 - acc:
0.9452 - precision: 0.9513 - recall: 0.9422 - val_loss: 0.0202 - val_acc: 0.9
933 - val_precision: 0.9939 - val_recall: 0.9933
Epoch 99/100
137/137 [==============================] - 630s 5s/step - loss: 0.1678 - acc:
0.9441 - precision: 0.9479 - recall: 0.9403 - val_loss: 0.0254 - val_acc: 0.9
926 - val_precision: 0.9932 - val_recall: 0.9919
Epoch 100/100
137/137 [==============================] - 576s 4s/step - loss: 0.1480 - acc:
```

```
0.9512 - precision: 0.9565 - recall: 0.9484 - val_loss: 0.0756 - val_acc: 0.9
742 - val_precision: 0.9742 - val_recall: 0.9742
```

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_epoc
h)
predicted_classes = np.argmax(predictions, axis=1)
```

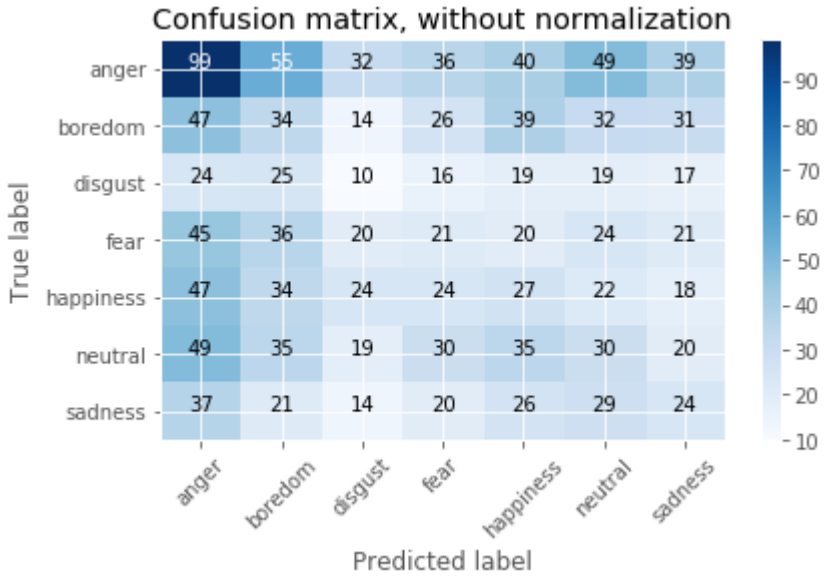
In [6]:
```
true_classes = test_set.classes
class_labels = list(test_set.class_indices.keys())
```

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

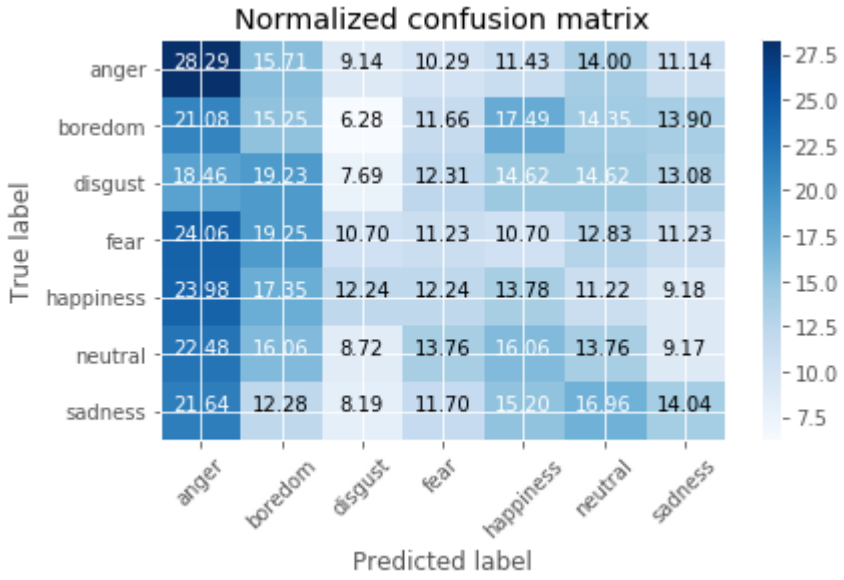
```
             precision    recall  f1-score   support

      anger       0.28      0.28      0.28       350
    boredom       0.14      0.15      0.15       223
    disgust       0.08      0.08      0.08       130
       fear       0.12      0.11      0.12       187
  happiness       0.13      0.14      0.13       196
    neutral       0.15      0.14      0.14       218
    sadness       0.14      0.14      0.14       171

avg / total       0.17      0.17      0.17      1475
```

In [11]:
```
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_lstm.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_lstm.png")
plt.show()
```

```
Confusion matrix, without normalization
[[99 55 32 36 40 49 39]
 [47 34 14 26 39 32 31]
 [24 25 10 16 19 19 17]
 [45 36 20 21 20 24 21]
 [47 34 24 24 27 22 18]
 [49 35 19 30 35 30 20]
 [37 21 14 20 26 29 24]]
```




```
Normalized confusion matrix
[[28.2857 15.7143  9.1429 10.2857 11.4286 14.     11.1429]
 [21.0762 15.2466  6.278  11.6592 17.4888 14.3498 13.9013]
 [18.4615 19.2308  7.6923 12.3077 14.6154 14.6154 13.0769]
 [24.0642 19.2513 10.6952 11.2299 10.6952 12.8342 11.2299]
 [23.9796 17.3469 12.2449 12.2449 13.7755 11.2245  9.1837]
 [22.4771 16.055   8.7156 13.7615 16.055  13.7615  9.1743]
 [21.6374 12.2807  8.1871 11.6959 15.2047 16.9591 14.0351]]
```



In [12]:
```
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_lstm.png")
```