In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
import glob
from sklearn.model_selection import train_test_split
from keras.layers import Dropout, Dense
from keras.layers.normalization import BatchNormalization
from keras.models import Sequential, load_model
from keras.applications import VGG16
from sklearn.metrics import accuracy_score, confusion_matrix
```

```
C:\Users\LENOVO\Anaconda3\lib\site-packages\h5py\__init__.py:36: Future
Warning: Conversion of the second argument of issubdtype from `float` t
o `np.floating` is deprecated. In future, it will be treated as `np.flo
at64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

In [2]:

```
def load_fer2013():
    data = pd.read_csv('fer2013.csv')
    pixels = data['pixels'].tolist()
    width,height = 48,48
    faces = []
    input_shape = (48,48,1)
    for sequence in pixels:
        face = [int(pixel) for pixel in sequence.split(' ')]
        face = np.asarray(face).reshape(width,height)
        face=  cv2.resize(face.astype('uint8'), input_shape[:2])
        faces.append(face.astype('float32'))
    faces = np.asarray(faces)
    faces = np.expand_dims(faces,-1)
    emotions = pd.get_dummies(data['emotion']).as_matrix()
    return faces, emotions
```

In [3]:

```
def split_data(x, y, validation_split=.2):

    num_samples = len(x)
    num_train_samples = int((1 - validation_split)*num_samples)
    train_x = x[:num_train_samples]
    train_y = y[:num_train_samples]
    val_x = x[num_train_samples:]
    val_y = y[num_train_samples:]
    train_data = (train_x, train_y)
    val_data = (val_x, val_y)
    return train_data, val_data
```

In [4]:

```
from keras.layers import Activation, Convolution2D, Dropout, Conv2D
from keras.layers import AveragePooling2D, BatchNormalization
from keras.layers import GlobalAveragePooling2D
from keras.models import Sequential
from keras.layers import Flatten
from keras.models import Model
from keras.layers import Input
from keras.layers import MaxPooling2D
from keras.layers import SeparableConv2D
from keras import layers
from keras.regularizers import l2
```

In [6]:

```
def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255.0
    if v2:
        x = x - 0.5
        x = x * 2.0
    return x
```

In [7]:

```
def mini_XCEPTION(input_shape, num_classes, l2_regularization=0.01):
    regularization = l2(l2_regularization)
    # base
    img_input = Input(input_shape)
    x = Conv2D(8, (3, 3), strides=(1, 1), kernel_regularizer=regularization,
               use_bias=False)(img_input)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Conv2D(8, (3, 3), strides=(1, 1), kernel_regularizer=regularization,
               use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    # module 1
    residual = Conv2D(16, (1, 1), strides=(2, 2),
                      padding='same', use_bias=False)(x)
    residual = BatchNormalization()(residual)

    x = SeparableConv2D(16, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = SeparableConv2D(16, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)

    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    x = layers.add([x, residual])

    # module 2
    residual = Conv2D(32, (1, 1), strides=(2, 2),
                      padding='same', use_bias=False)(x)
    residual = BatchNormalization()(residual)

    x = SeparableConv2D(32, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = SeparableConv2D(32, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)

    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    x = layers.add([x, residual])

    # module 3
    residual = Conv2D(64, (1, 1), strides=(2, 2),
                      padding='same', use_bias=False)(x)
    residual = BatchNormalization()(residual)

    x = SeparableConv2D(64, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = SeparableConv2D(64, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)

    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    x = layers.add([x, residual])

    # module 4
    residual = Conv2D(128, (1, 1), strides=(2, 2),
                      padding='same', use_bias=False)(x)
    residual = BatchNormalization()(residual)

    x = SeparableConv2D(128, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = SeparableConv2D(128, (3, 3), padding='same',
                        kernel_regularizer=regularization,
                        use_bias=False)(x)
    x = BatchNormalization()(x)

    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    x = layers.add([x, residual])

    x = Conv2D(num_classes, (3, 3),
               # kernel_regularizer=regularization,
               padding='same')(x)
    x = GlobalAveragePooling2D()(x)
    output = Activation('softmax', name='predictions')(x)

    model = Model(img_input, output)
    return model
```

In [2]:

```
import cv2
from keras import applications
from keras import optimizers
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten , BatchNormalization
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
from keras.layers import Input
batch_size = 128
num_classes = 10
epochs = 12
```

In [9]:

```
datagen = ImageDataGenerator(featurewise_center=False,
                        featurewise_std_normalization=True,
                        rotation_range=10,
                        width_shift_range=0.1,
                        height_shift_range=0.1,
                        zoom_range=.1,
                        horizontal_flip=True)
```

```
/usr/local/lib/python3.5/dist-packages/keras_preprocessing/image/image_
data_generator.py:346: UserWarning: This ImageDataGenerator specifies `
```

```
featurewise_std_normalization`, which overrides setting of `featurewise
_center`.
  warnings.warn('This ImageDataGenerator specifies '
```

In [4]:

```
from keras.regularizers import l2
from keras.layers import Activation, Convolution2D, Dropout, Conv2D
from keras.layers import AveragePooling2D, BatchNormalization
from keras.layers import GlobalAveragePooling2D
from keras.models import Sequential
from keras.layers import Flatten
from keras.models import Model
from keras.layers import Input
from keras.layers import MaxPooling2D
from keras.layers import SeparableConv2D
from keras import layers
```

In [11]:

```
input_shape =(48,48,1)
model  = mini_XCEPTION(input_shape,7)
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
```

```
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorfl
ow/python/framework/op_def_library.py:263: colocate_with (from tensorfl
ow.python.framework.ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Colocations handled automatically by placer.
_______________________________________________________________________
___________________________
Layer (type)                    Output Shape         Param #     Connec
ted to
=======================================================================
===========================
input_1 (InputLayer)            (None, 48, 48, 1)    0

_______________________________________________________________________
___________________________
conv2d_1 (Conv2D)               (None, 46, 46, 8)    72          input_
```

```
1[0][0]
_______________________________________________________________________
___________________________
batch_normalization_1 (BatchNor (None, 46, 46, 8)    32          conv2d
_1[0][0]
_______________________________________________________________________
___________________________
activation_1 (Activation)       (None, 46, 46, 8)    0           batch_
normalization_1[0][0]
_______________________________________________________________________
___________________________
conv2d_2 (Conv2D)               (None, 44, 44, 8)    576         activa
tion_1[0][0]
_______________________________________________________________________
___________________________
batch_normalization_2 (BatchNor (None, 44, 44, 8)    32          conv2d
_2[0][0]
_______________________________________________________________________
___________________________
activation_2 (Activation)       (None, 44, 44, 8)    0           batch_
normalization_2[0][0]
_______________________________________________________________________
___________________________
separable_conv2d_1 (SeparableCo (None, 44, 44, 16)   200         activa
tion_2[0][0]
_______________________________________________________________________
___________________________
batch_normalization_4 (BatchNor (None, 44, 44, 16)   64          separa
ble_conv2d_1[0][0]
_______________________________________________________________________
___________________________
activation_3 (Activation)       (None, 44, 44, 16)   0           batch_
normalization_4[0][0]
_______________________________________________________________________
___________________________
separable_conv2d_2 (SeparableCo (None, 44, 44, 16)   400         activa
tion_3[0][0]
_______________________________________________________________________
___________________________
```

```
batch_normalization_5 (BatchNor (None, 44, 44, 16)   64          separa
ble_conv2d_2[0][0]
_______________________________________________________________________
___________________________
conv2d_3 (Conv2D)               (None, 22, 22, 16)   128         activa
tion_2[0][0]
_______________________________________________________________________
___________________________
max_pooling2d_1 (MaxPooling2D)  (None, 22, 22, 16)   0           batch_
normalization_5[0][0]
_______________________________________________________________________
___________________________
batch_normalization_3 (BatchNor (None, 22, 22, 16)   64          conv2d
_3[0][0]
_______________________________________________________________________
___________________________
add_1 (Add)                     (None, 22, 22, 16)   0           max_po
oling2d_1[0][0]
                                                                 batch_
normalization_3[0][0]
_______________________________________________________________________
___________________________
separable_conv2d_3 (SeparableCo (None, 22, 22, 32)   656         add_1
[0][0]
_______________________________________________________________________
___________________________
batch_normalization_7 (BatchNor (None, 22, 22, 32)   128         separa
ble_conv2d_3[0][0]
_______________________________________________________________________
___________________________
activation_4 (Activation)       (None, 22, 22, 32)   0           batch_
normalization_7[0][0]
_______________________________________________________________________
___________________________
separable_conv2d_4 (SeparableCo (None, 22, 22, 32)   1312        activa
tion_4[0][0]
_______________________________________________________________________
___________________________
batch_normalization_8 (BatchNor (None, 22, 22, 32)   128         separa
ble_conv2d_4[0][0]

_______________________________________________________________________
___________________________
conv2d_4 (Conv2D)               (None, 11, 11, 32)   512         add_1
[0][0]

_______________________________________________________________________
___________________________
max_pooling2d_2 (MaxPooling2D)  (None, 11, 11, 32)   0           batch_
normalization_8[0][0]

_______________________________________________________________________
___________________________
batch_normalization_6 (BatchNor (None, 11, 11, 32)   128         conv2d
_4[0][0]

_______________________________________________________________________
___________________________
add_2 (Add)                     (None, 11, 11, 32)   0           max_po
oling2d_2[0][0]
                                                                 batch_
normalization_6[0][0]

_______________________________________________________________________
___________________________
separable_conv2d_5 (SeparableCo (None, 11, 11, 64)   2336        add_2
[0][0]

_______________________________________________________________________
___________________________
batch_normalization_10 (BatchNo (None, 11, 11, 64)   256         separa
ble_conv2d_5[0][0]

_______________________________________________________________________
___________________________
activation_5 (Activation)       (None, 11, 11, 64)   0           batch_
normalization_10[0][0]

_______________________________________________________________________
___________________________
separable_conv2d_6 (SeparableCo (None, 11, 11, 64)   4672        activa
tion_5[0][0]

_______________________________________________________________________
___________________________
batch_normalization_11 (BatchNo (None, 11, 11, 64)   256         separa
ble_conv2d_6[0][0]
```

```
_______________________________________________________________________
___________________________
conv2d_5 (Conv2D)               (None, 6, 6, 64)     2048        add_2
[0][0]
_______________________________________________________________________
___________________________
max_pooling2d_3 (MaxPooling2D)  (None, 6, 6, 64)     0           batch_
normalization_11[0][0]
_______________________________________________________________________
___________________________
batch_normalization_9 (BatchNor (None, 6, 6, 64)     256         conv2d
_5[0][0]
_______________________________________________________________________
___________________________
add_3 (Add)                     (None, 6, 6, 64)     0           max_po
oling2d_3[0][0]
                                                                 batch_
normalization_9[0][0]
_______________________________________________________________________
___________________________
separable_conv2d_7 (SeparableCo (None, 6, 6, 128)    8768        add_3
[0][0]
_______________________________________________________________________
___________________________
batch_normalization_13 (BatchNo (None, 6, 6, 128)    512         separa
ble_conv2d_7[0][0]
_______________________________________________________________________
___________________________
activation_6 (Activation)       (None, 6, 6, 128)    0           batch_
normalization_13[0][0]
_______________________________________________________________________
___________________________
separable_conv2d_8 (SeparableCo (None, 6, 6, 128)    17536       activa
tion_6[0][0]
_______________________________________________________________________
___________________________
batch_normalization_14 (BatchNo (None, 6, 6, 128)    512         separa
ble_conv2d_8[0][0]
_______________________________________________________________________
```

```
___________________________
conv2d_6 (Conv2D)               (None, 3, 3, 128)    8192        add_3
[0][0]

_______________________________________________________________________
___________________________
max_pooling2d_4 (MaxPooling2D)  (None, 3, 3, 128)    0           batch_
normalization_14[0][0]

_______________________________________________________________________
___________________________
batch_normalization_12 (BatchNo (None, 3, 3, 128)    512         conv2d
_6[0][0]

_______________________________________________________________________
___________________________
add_4 (Add)                     (None, 3, 3, 128)    0           max_po
oling2d_4[0][0]
                                                                 batch_
normalization_12[0][0]

_______________________________________________________________________
___________________________
conv2d_7 (Conv2D)               (None, 3, 3, 7)      8071        add_4
[0][0]

_______________________________________________________________________
___________________________
global_average_pooling2d_1 (Glo (None, 7)            0           conv2d
_7[0][0]

_______________________________________________________________________
___________________________
predictions (Activation)        (None, 7)            0           global
_average_pooling2d_1[0][0]
=======================================================================
===========================
Total params: 58,423
Trainable params: 56,951
Non-trainable params: 1,472

_______________________________________________________________________
___________________________
```

In [41]:

```
faces , emotions = load_fer2013()
```

```
/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:14: Future
Warning: Method .as_matrix will be removed in a future version. Use .va
lues instead.
```

In [42]:

```
faces = preprocess_input(faces)
```

In [43]:

```
num_shapes , num_classes = emotions.shape
```

In [44]:

```
num_shapes
```

Out[44]:

```
35887
```

In [45]:

```
num_classes
```

Out[45]:

```
7
```

In [46]:

```
validation_split = 0.2
```

In [47]:

```
train_data, val_data = split_data(faces, emotions, validation_split)
```

In [48]:

```
train_faces , train_emotion = train_data
```

In [62]:

```
batch_size = 64
num_epochs = 100
input_shape = (48, 48, 1)
validation_split = .2
verbose = 1
num_classes = 7
patience = 50
```

In [ ]:

```
base_path = emodetect/
```

In [ ]:

```
log_file_path = base_path + dataset_name + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)
```

```
early_stop = EarlyStopping('val_loss', patience=patience)
reduce_lr = ReduceLROnPlateau('val_loss', factor=0.1,
                              patience=int(patience/4), verbose=1)
trained_models_path = base_path + dataset_name + '_mini_XCEPTION'
model_names = trained_models_path + '.{epoch:02d}-{val_acc:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(model_names, 'val_loss', verbose=1,
                                   save_best_only=True
)
callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]
```

In [63]:

```
model.fit_generator(datagen.flow(train_faces, train_emotion,
                                 batch_size),
                    steps_per_epoch=100,
                    epochs=num_epochs, verbose=1,
                    validation_data=val_data)
```

```
Epoch 1/100
100/100 [==============================] - 124s 1s/step - loss: 1.1164
- acc: 0.5858 - val_loss: 1.1896 - val_acc: 0.5631
Epoch 2/100
100/100 [==============================] - 130s 1s/step - loss: 1.1310
- acc: 0.5744 - val_loss: 1.1983 - val_acc: 0.5612
Epoch 3/100
100/100 [==============================] - 132s 1s/step - loss: 1.1097
- acc: 0.5786 - val_loss: 1.4524 - val_acc: 0.4979
Epoch 4/100
100/100 [==============================] - 131s 1s/step - loss: 1.1180
- acc: 0.5792 - val_loss: 1.2078 - val_acc: 0.5598
Epoch 5/100
100/100 [==============================] - 131s 1s/step - loss: 1.1085
- acc: 0.5770 - val_loss: 1.1673 - val_acc: 0.5609
Epoch 6/100
100/100 [==============================] - 136s 1s/step - loss: 1.0983
- acc: 0.5894 - val_loss: 1.1634 - val_acc: 0.5667
Epoch 7/100
100/100 [==============================] - 134s 1s/step - loss: 1.1093
- acc: 0.5864 - val_loss: 1.2080 - val_acc: 0.5532
Epoch 8/100
100/100 [==============================] - 131s 1s/step - loss: 1.1135
```

```
- acc: 0.5909 - val_loss: 1.2182 - val_acc: 0.5481
Epoch 9/100
100/100 [==============================] - 138s 1s/step - loss: 1.1238
- acc: 0.5785 - val_loss: 1.1362 - val_acc: 0.5790
Epoch 10/100
100/100 [==============================] - 126s 1s/step - loss: 1.0867
- acc: 0.5933 - val_loss: 1.1582 - val_acc: 0.5766
Epoch 11/100
100/100 [==============================] - 128s 1s/step - loss: 1.0924
- acc: 0.5958 - val_loss: 1.1772 - val_acc: 0.5734
Epoch 12/100
100/100 [==============================] - 132s 1s/step - loss: 1.0963
- acc: 0.5861 - val_loss: 1.2263 - val_acc: 0.5639
Epoch 13/100
100/100 [==============================] - 128s 1s/step - loss: 1.1001
- acc: 0.5841 - val_loss: 1.2141 - val_acc: 0.5716
Epoch 14/100
100/100 [==============================] - 132s 1s/step - loss: 1.0762
- acc: 0.6001 - val_loss: 1.1916 - val_acc: 0.5549
Epoch 15/100
100/100 [==============================] - 130s 1s/step - loss: 1.0752
- acc: 0.6041 - val_loss: 1.2195 - val_acc: 0.5502
Epoch 16/100
100/100 [==============================] - 135s 1s/step - loss: 1.0858
- acc: 0.5959 - val_loss: 1.1558 - val_acc: 0.5677
Epoch 17/100
100/100 [==============================] - 134s 1s/step - loss: 1.1078
- acc: 0.5872 - val_loss: 1.1293 - val_acc: 0.5840
Epoch 18/100
100/100 [==============================] - 126s 1s/step - loss: 1.0925
- acc: 0.5878 - val_loss: 1.1434 - val_acc: 0.5692
Epoch 19/100
100/100 [==============================] - 135s 1s/step - loss: 1.0994
- acc: 0.5939 - val_loss: 1.1655 - val_acc: 0.5678
Epoch 20/100
100/100 [==============================] - 124s 1s/step - loss: 1.0852
- acc: 0.5936 - val_loss: 1.1189 - val_acc: 0.5837
Epoch 21/100
100/100 [==============================] - 125s 1s/step - loss: 1.0815
```

```
- acc: 0.5961 - val_loss: 1.1045 - val_acc: 0.5900
Epoch 22/100
100/100 [==============================] - 130s 1s/step - loss: 1.0698
- acc: 0.6030 - val_loss: 1.2405 - val_acc: 0.5450
Epoch 23/100
100/100 [==============================] - 134s 1s/step - loss: 1.0636
- acc: 0.6019 - val_loss: 1.1557 - val_acc: 0.5768
Epoch 24/100
100/100 [==============================] - 135s 1s/step - loss: 1.0696
- acc: 0.5992 - val_loss: 1.1591 - val_acc: 0.5692
Epoch 25/100
100/100 [==============================] - 132s 1s/step - loss: 1.0765
- acc: 0.5994 - val_loss: 1.1061 - val_acc: 0.5931
Epoch 26/100
100/100 [==============================] - 135s 1s/step - loss: 1.0796
- acc: 0.5925 - val_loss: 1.1694 - val_acc: 0.5699
Epoch 27/100
100/100 [==============================] - 128s 1s/step - loss: 1.0711
- acc: 0.5918 - val_loss: 1.1387 - val_acc: 0.5821
Epoch 28/100
100/100 [==============================] - 126s 1s/step - loss: 1.0563
- acc: 0.6067 - val_loss: 1.2104 - val_acc: 0.5699
Epoch 29/100
100/100 [==============================] - 124s 1s/step - loss: 1.0613
- acc: 0.6031 - val_loss: 1.1490 - val_acc: 0.5711
Epoch 30/100
100/100 [==============================] - 129s 1s/step - loss: 1.0623
- acc: 0.6025 - val_loss: 1.1649 - val_acc: 0.5740
Epoch 31/100
100/100 [==============================] - 124s 1s/step - loss: 1.0774
- acc: 0.6020 - val_loss: 1.2368 - val_acc: 0.5626
Epoch 32/100
100/100 [==============================] - 123s 1s/step - loss: 1.0764
- acc: 0.5894 - val_loss: 1.1174 - val_acc: 0.5894
Epoch 33/100
100/100 [==============================] - 132s 1s/step - loss: 1.0566
- acc: 0.5947 - val_loss: 1.1247 - val_acc: 0.5794
Epoch 34/100
100/100 [==============================] - 128s 1s/step - loss: 1.0592
```

```
- acc: 0.6053 - val_loss: 1.1190 - val_acc: 0.5770
Epoch 35/100
100/100 [==============================] - 119s 1s/step - loss: 1.0712
- acc: 0.6006 - val_loss: 1.0966 - val_acc: 0.6004
Epoch 36/100
100/100 [==============================] - 132s 1s/step - loss: 1.0469
- acc: 0.6147 - val_loss: 1.1326 - val_acc: 0.5848
Epoch 37/100
100/100 [==============================] - 124s 1s/step - loss: 1.0448
- acc: 0.6058 - val_loss: 1.1263 - val_acc: 0.5901
Epoch 38/100
100/100 [==============================] - 129s 1s/step - loss: 1.0511
- acc: 0.6034 - val_loss: 1.1153 - val_acc: 0.5843
Epoch 39/100
100/100 [==============================] - 125s 1s/step - loss: 1.0575
- acc: 0.5980 - val_loss: 1.0999 - val_acc: 0.5970
Epoch 40/100
100/100 [==============================] - 125s 1s/step - loss: 1.0457
- acc: 0.6078 - val_loss: 1.1879 - val_acc: 0.5658
Epoch 41/100
100/100 [==============================] - 133s 1s/step - loss: 1.0277
- acc: 0.6189 - val_loss: 1.1920 - val_acc: 0.5724
Epoch 42/100
100/100 [==============================] - 133s 1s/step - loss: 1.0410
- acc: 0.6134 - val_loss: 1.1571 - val_acc: 0.5722
Epoch 43/100
100/100 [==============================] - 138s 1s/step - loss: 1.0740
- acc: 0.5972 - val_loss: 1.0916 - val_acc: 0.5949
Epoch 44/100
100/100 [==============================] - 123s 1s/step - loss: 1.0367
- acc: 0.6070 - val_loss: 1.1637 - val_acc: 0.5772
Epoch 45/100
100/100 [==============================] - 125s 1s/step - loss: 1.0742
- acc: 0.6065 - val_loss: 1.1859 - val_acc: 0.5726
Epoch 46/100
100/100 [==============================] - 139s 1s/step - loss: 1.0322
- acc: 0.6080 - val_loss: 1.1544 - val_acc: 0.5762
Epoch 47/100
100/100 [==============================] - 123s 1s/step - loss: 1.0675
```

```
- acc: 0.5991 - val_loss: 1.1277 - val_acc: 0.5836
Epoch 48/100
100/100 [==============================] - 134s 1s/step - loss: 1.0312
- acc: 0.6164 - val_loss: 1.0863 - val_acc: 0.6021
Epoch 49/100
100/100 [==============================] - 129s 1s/step - loss: 1.0334
- acc: 0.6097 - val_loss: 1.1329 - val_acc: 0.5780
Epoch 50/100
100/100 [==============================] - 127s 1s/step - loss: 1.0554
- acc: 0.5976 - val_loss: 1.1328 - val_acc: 0.5784
Epoch 51/100
100/100 [==============================] - 129s 1s/step - loss: 1.0256
- acc: 0.6155 - val_loss: 1.2352 - val_acc: 0.5368
Epoch 52/100
100/100 [==============================] - 128s 1s/step - loss: 1.0480
- acc: 0.6036 - val_loss: 1.0989 - val_acc: 0.5942
Epoch 53/100
100/100 [==============================] - 139s 1s/step - loss: 1.0350
- acc: 0.6131 - val_loss: 1.1137 - val_acc: 0.5992
Epoch 54/100
100/100 [==============================] - 127s 1s/step - loss: 1.0361
- acc: 0.6164 - val_loss: 1.1646 - val_acc: 0.5719
Epoch 55/100
100/100 [==============================] - 132s 1s/step - loss: 1.0414
- acc: 0.6072 - val_loss: 1.1049 - val_acc: 0.5961
Epoch 56/100
100/100 [==============================] - 132s 1s/step - loss: 1.0377
- acc: 0.6155 - val_loss: 1.1591 - val_acc: 0.5772
Epoch 57/100
100/100 [==============================] - 135s 1s/step - loss: 1.0172
- acc: 0.6259 - val_loss: 1.0844 - val_acc: 0.6071
Epoch 58/100
100/100 [==============================] - 132s 1s/step - loss: 1.0336
- acc: 0.6148 - val_loss: 1.0802 - val_acc: 0.6042
Epoch 59/100
100/100 [==============================] - 130s 1s/step - loss: 1.0016
- acc: 0.6280 - val_loss: 1.1060 - val_acc: 0.5970
Epoch 60/100
100/100 [==============================] - 133s 1s/step - loss: 1.0369
```

```
- acc: 0.6069 - val_loss: 1.0735 - val_acc: 0.6082
Epoch 61/100
100/100 [==============================] - 130s 1s/step - loss: 1.0197
- acc: 0.6205 - val_loss: 1.0858 - val_acc: 0.6045
Epoch 62/100
100/100 [==============================] - 132s 1s/step - loss: 1.0262
- acc: 0.6186 - val_loss: 1.0660 - val_acc: 0.6096
Epoch 63/100
100/100 [==============================] - 139s 1s/step - loss: 1.0212
- acc: 0.6182 - val_loss: 1.1269 - val_acc: 0.5874
Epoch 64/100
100/100 [==============================] - 129s 1s/step - loss: 1.0183
- acc: 0.6159 - val_loss: 1.1380 - val_acc: 0.5864
Epoch 65/100
100/100 [==============================] - 131s 1s/step - loss: 1.0237
- acc: 0.6208 - val_loss: 1.0897 - val_acc: 0.6025
Epoch 66/100
100/100 [==============================] - 131s 1s/step - loss: 1.0310
- acc: 0.6120 - val_loss: 1.1106 - val_acc: 0.5977
Epoch 67/100
100/100 [==============================] - 135s 1s/step - loss: 1.0180
- acc: 0.6248 - val_loss: 1.0779 - val_acc: 0.6080
Epoch 68/100
100/100 [==============================] - 135s 1s/step - loss: 1.0133
- acc: 0.6235 - val_loss: 1.1081 - val_acc: 0.5971
Epoch 69/100
100/100 [==============================] - 135s 1s/step - loss: 1.0031
- acc: 0.6270 - val_loss: 1.0848 - val_acc: 0.5942
Epoch 70/100
100/100 [==============================] - 129s 1s/step - loss: 1.0275
- acc: 0.6108 - val_loss: 1.0931 - val_acc: 0.5986
Epoch 71/100
100/100 [==============================] - 132s 1s/step - loss: 1.0303
- acc: 0.6194 - val_loss: 1.1369 - val_acc: 0.5897
Epoch 72/100
100/100 [==============================] - 135s 1s/step - loss: 1.0259
- acc: 0.6196 - val_loss: 1.1715 - val_acc: 0.5740
Epoch 73/100
100/100 [==============================] - 133s 1s/step - loss: 1.0111
```

```
- acc: 0.6300 - val_loss: 1.2282 - val_acc: 0.5588
Epoch 74/100
100/100 [==============================] - 128s 1s/step - loss: 1.0110
- acc: 0.6241 - val_loss: 1.1039 - val_acc: 0.5876
Epoch 75/100
100/100 [==============================] - 133s 1s/step - loss: 0.9988
- acc: 0.6267 - val_loss: 1.1469 - val_acc: 0.5901
Epoch 76/100
100/100 [==============================] - 130s 1s/step - loss: 1.0175
- acc: 0.6213 - val_loss: 1.1153 - val_acc: 0.5911
Epoch 77/100
100/100 [==============================] - 134s 1s/step - loss: 0.9945
- acc: 0.6279 - val_loss: 1.1104 - val_acc: 0.6060
Epoch 78/100
100/100 [==============================] - 136s 1s/step - loss: 0.9960
- acc: 0.6314 - val_loss: 1.1223 - val_acc: 0.5997
Epoch 79/100
100/100 [==============================] - 131s 1s/step - loss: 1.0016
- acc: 0.6291 - val_loss: 1.0870 - val_acc: 0.6018
Epoch 80/100
100/100 [==============================] - 129s 1s/step - loss: 1.0257
- acc: 0.6211 - val_loss: 1.0927 - val_acc: 0.6032
Epoch 81/100
100/100 [==============================] - 133s 1s/step - loss: 1.0097
- acc: 0.6300 - val_loss: 1.0935 - val_acc: 0.6073
Epoch 82/100
100/100 [==============================] - 130s 1s/step - loss: 1.0166
- acc: 0.6205 - val_loss: 1.1970 - val_acc: 0.5690
Epoch 83/100
100/100 [==============================] - 134s 1s/step - loss: 0.9997
- acc: 0.6259 - val_loss: 1.1039 - val_acc: 0.5985
Epoch 84/100
100/100 [==============================] - 137s 1s/step - loss: 0.9910
- acc: 0.6286 - val_loss: 1.1052 - val_acc: 0.5978
Epoch 85/100
100/100 [==============================] - 130s 1s/step - loss: 1.0241
- acc: 0.6162 - val_loss: 1.1084 - val_acc: 0.5919
Epoch 86/100
100/100 [==============================] - 134s 1s/step - loss: 0.9999
```

```
- acc: 0.6239 - val_loss: 1.1141 - val_acc: 0.5918
Epoch 87/100
100/100 [==============================] - 129s 1s/step - loss: 1.0036
- acc: 0.6280 - val_loss: 1.2237 - val_acc: 0.5627
Epoch 88/100
100/100 [==============================] - 122s 1s/step - loss: 1.0127
- acc: 0.6209 - val_loss: 1.0957 - val_acc: 0.5947
Epoch 89/100
100/100 [==============================] - 134s 1s/step - loss: 0.9928
- acc: 0.6247 - val_loss: 1.2730 - val_acc: 0.5364
Epoch 90/100
100/100 [==============================] - 127s 1s/step - loss: 1.0101
- acc: 0.6219 - val_loss: 1.1031 - val_acc: 0.6027
Epoch 91/100
100/100 [==============================] - 132s 1s/step - loss: 0.9739
- acc: 0.6395 - val_loss: 1.0707 - val_acc: 0.6116
Epoch 92/100
100/100 [==============================] - 130s 1s/step - loss: 1.0076
- acc: 0.6208 - val_loss: 1.0796 - val_acc: 0.5979
Epoch 93/100
100/100 [==============================] - 127s 1s/step - loss: 0.9767
- acc: 0.6323 - val_loss: 1.0758 - val_acc: 0.6042
Epoch 94/100
100/100 [==============================] - 133s 1s/step - loss: 0.9980
- acc: 0.6230 - val_loss: 1.1292 - val_acc: 0.5938
Epoch 95/100
100/100 [==============================] - 137s 1s/step - loss: 0.9988
- acc: 0.6347 - val_loss: 1.0638 - val_acc: 0.6120
Epoch 96/100
100/100 [==============================] - 133s 1s/step - loss: 0.9965
- acc: 0.6255 - val_loss: 1.0827 - val_acc: 0.6070
Epoch 97/100
100/100 [==============================] - 130s 1s/step - loss: 0.9851
- acc: 0.6258 - val_loss: 1.0969 - val_acc: 0.5940
Epoch 98/100
100/100 [==============================] - 132s 1s/step - loss: 0.9919
- acc: 0.6325 - val_loss: 1.1130 - val_acc: 0.5993
Epoch 99/100
100/100 [==============================] - 126s 1s/step - loss: 1.0046
```

```
- acc: 0.6257 - val_loss: 1.1689 - val_acc: 0.5651
Epoch 100/100
100/100 [==============================] - 135s 1s/step - loss: 0.9843
- acc: 0.6280 - val_loss: 1.0550 - val_acc: 0.6089
```

Out[63]:
```
<keras.callbacks.History at 0x7ff52e3f7470>
```

In [64]:
```
model.save('mini_xception_model.hdf5')
```

In [16]:
```
emotion_classifier.history
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-16-ed003721168f> in <module>()
----> 1 emotion_classifier.history

AttributeError: 'Model' object has no attribute 'history'
```

In [ ]:
```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np

# parameters for loading data and images
detection_model_path = 'haarcascade_frontalface_default.xml'
emotion_model_path = 'mini_xception_model1.hdf5'

# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad", "surprised",
 "neutral"]


# starting video streaming
```

```
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
while True:
    frame = camera.read()[1]
    #reading the frame
    frame = imutils.resize(frame,width=400)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_detection.detectMultiScale(gray,scaleFactor=1.1,minNei
ghbors=5,minSize=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)

    canvas = np.zeros((250, 300, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                    # Extract the ROI of the face from the grayscale im
age, resize it to a fixed 48x48 pixels, and then prepare
            # the ROI for classification via the CNN
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (48, 48))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)


        preds = emotion_classifier.predict(roi)[0]
        emotion_probability = np.max(preds)
        label = EMOTIONS[preds.argmax()]


    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
                # construct the label text
                text = "{}: {:.2f}%".format(emotion, prob * 100)
                w = int(prob * 300)
                cv2.rectangle(canvas, (7, (i * 35) + 5),
                (w, (i * 35) + 35), (0, 0, 255), -1)
                cv2.putText(canvas, text, (10, (i * 35) + 23),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45,
                (255, 255, 255), 2)
                cv2.putText(frameClone, label, (fX, fY - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
                cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),
                              (0, 0, 255), 2)

    cv2.imshow('your_face', frameClone)
    cv2.imshow("Probabilities", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

camera.release()
cv2.destroyAllWindows()
```

In [5]:

```
!pip install --upgrade imutils
```

```
Collecting imutils
  Downloading https://files.pythonhosted.org/packages/5e/0c/659c2bdae8e
8ca5ef810b9da02db28feaa29ea448ff36b65a1664ff28142/imutils-0.5.2.tar.gz
Building wheels for collected packages: imutils
  Running setup.py bdist_wheel for imutils: started
  Running setup.py bdist_wheel for imutils: finished with status 'done'
  Stored in directory: C:\Users\LENOVO\AppData\Local\pip\Cache\wheels\b
2\40\59\139d450e68847ef2f27d876d527b13389dac23df0f66526b5d
Successfully built imutils
Installing collected packages: imutils
Successfully installed imutils-0.5.2
```

```
distributed 1.21.8 requires msgpack, which is not installed.
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade
pip' command.
```