Importing Modules ¶

In [1]:

```
import numpy as np

from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D
from keras.optimizers import Adam
from keras.layers import MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator
```

Loading Dataset

In [2]:

```
train_dir = '/content/train'
test_dir = '../content/test'

train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_data = train_datagen.flow_from_directory(
                train_dir,
                target_size=(48,48),
                batch_size=64,
                color_mode="grayscale",
                class_mode='categorical')

test_data = test_datagen.flow_from_directory(
                test_dir,
                target_size=(48,48),
                batch_size=64,
                color_mode="grayscale",

    class_mode='categorical')
```

```
Found 28709 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

Model Creation

In [3]:

```
emotion_model = Sequential()
emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))
emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))
```

Model Compile, Run and Testing

In [4]:

```
emotion_model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001,
emotion_model_info = emotion_model.fit_generator(
        train_data,
        steps_per_epoch=28709 // 64,
        epochs=50,
        validation_data=test_data,
        validation_steps=7178 // 64,
)
```

```
Epoch 36/50
448/448 [==============================] - 9s 21ms/step - loss: 0.6352 - accurac
y: 0.7704 - val_loss: 1.0899 - val_accuracy: 0.6226
Epoch 37/50
448/448 [==============================] - 9s 21ms/step - loss: 0.6151 - accurac
y: 0.7764 - val_loss: 1.1003 - val_accuracy: 0.6189
Epoch 38/50
448/448 [==============================] - 9s 21ms/step - loss: 0.5891 - accurac
y: 0.7846 - val_loss: 1.1090 - val_accuracy: 0.6169
Epoch 39/50
448/448 [==============================] - 9s 21ms/step - loss: 0.5656 - accurac
y: 0.7955 - val_loss: 1.1097 - val_accuracy: 0.6175
Epoch 40/50
448/448 [==============================] - 9s 21ms/step - loss: 0.5433 - accurac
y: 0.8048 - val_loss: 1.1151 - val_accuracy: 0.6208
Epoch 41/50
448/448 [==============================] - 9s 21ms/step - loss: 0.5284 - accurac
y: 0.8070 - val_loss: 1.1356 - val_accuracy: 0.6201
Epoch 42/50
448/448 [==============================] - 10s 21ms/step - loss: 0.5156 - accurac
```

Model Accuracy: 0.6159

Saving Model (Weights, json file)

In [6]:

```
model_json = emotion_model.to_json()
with open("model.json", "w") as json_file:
    json_file.write(model_json)
    emotion_model.save_weights("model.h5")
    print("Saved model to disk")
```

```
Saved model to disk
```