In [1]:
```
from pyAudioAnalysis import audioTrainTest as aT
aT.featureAndTrain(["train/angry","train/calm","train/disgust","train/fearful"
,"train/happy","train/sad","train/surprised","train/neutral"], 1.0, 1.0, aT.sh
ortTermWindow, aT.shortTermStep, "gradientboosting", "svmSMtemp", False)
```

In [2]:
```
import os
import numpy as np
angry = []
for root, dirs, files in os.walk(r'total/test/angry/'):
    for file in files:
        if file.endswith('.wav'):
            angry.append(file)
calm = []
for root, dirs, files in os.walk(r'total/test/calm/'):
    for file in files:
        if file.endswith('.wav'):
            calm.append(file)
disgust = []
for root, dirs, files in os.walk(r'total/test/disgust/'):
    for file in files:
        if file.endswith('.wav'):
            disgust.append(file)
fearful = []
for root, dirs, files in os.walk(r'total/test/fearful/'):
    for file in files:
        if file.endswith('.wav'):
            fearful.append(file)
happy = []
for root, dirs, files in os.walk(r'total/test/happy/'):
    for file in files:
        if file.endswith('.wav'):
            happy.append(file)
sad = []
for root, dirs, files in os.walk(r'total/test/sad/'):
    for file in files:
        if file.endswith('.wav'):
            sad.append(file)
surprised = []
for root, dirs, files in os.walk(r'total/test/surprised/'):
    for file in files:
        if file.endswith('.wav'):
            surprised.append(file)
neutral = []
for root, dirs, files in os.walk(r'total/test/neutral/'):
    for file in files:
        if file.endswith('.wav'):
            neutral.append(file)
```

In [3]:
```
c = []
for i in angry:
    c = np.append(c,aT.fileClassification("total/test/angry/"+i, "svmSMtemp",
"gradientboosting"))
for i in calm:
    c = np.append(c,aT.fileClassification("total/test/calm/"+i, "svmSMtemp","g
radientboosting"))
for i in disgust:
    c = np.append(c,aT.fileClassification("total/test/disgust/"+i,"svmSMtemp",
"gradientboosting"))
for i in fearful:
    c = np.append(c,aT.fileClassification("total/test/fearful/"+i,"svmSMtemp",
"gradientboosting"))
for i in happy:
    c = np.append(c,aT.fileClassification("total/test/happy/"+i,"svmSMtemp","g
radientboosting"))
for i in sad:
    c = np.append(c,aT.fileClassification("total/test/sad/"+i,"svmSMtemp","gra
dientboosting"))
for i in surprised:
    c = np.append(c,aT.fileClassification("total/test/surprised/"+i,"svmSMtem
p","gradientboosting"))
for i in neutral:
    c = np.append(c,aT.fileClassification("total/test/neutral/"+i,"svmSMtemp",
"gradientboosting"))
c = np.reshape(c,(-1,8))
```

In [4]:
```
c
```

Out[4]:
```
array([[5.29879764e-02, 1.97444034e-02, 3.94672149e-01, ...,
        4.28417338e-02, 2.87867499e-01, 3.56616423e-03],
       [8.16039339e-01, 4.85471000e-03, 8.60645568e-03, ...,
        2.55223382e-02, 2.73949217e-02, 5.21422209e-03],
       [1.85357206e-01, 1.90329578e-03, 3.39633838e-03, ...,
        3.52874215e-03, 2.61378799e-03, 8.98657518e-05],
       ...,
       [2.34501017e-02, 1.04686808e-01, 1.84023212e-02, ...,
        9.13438759e-02, 6.40807691e-02, 4.38198992e-02],
       [9.85743099e-02, 1.56560903e-01, 8.75737535e-02, ...,
        7.29544465e-02, 4.65885375e-02, 3.79628820e-02],
       [1.91888203e-02, 2.17994795e-01, 3.43405430e-02, ...,
        7.46335180e-02, 1.56481520e-01, 2.15561577e-01]])
```

In [5]:
```
y_pred = np.argmax(c,axis = 1)
y_pred
```

Out[5]:
```
array([2, 0, 3, 0, 5, 0, 4, 0, 0, 0, 4, 0, 0, 6, 0, 0, 0, 3, 0, 0, 3, 0,
       2, 0, 0, 6, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 6, 0, 3, 0, 7,
       0, 0, 0, 0, 1, 1, 1, 5, 1, 4, 1, 1, 4, 1, 1, 3, 2, 1, 5, 1, 1, 1,
       1, 1, 1, 2, 6, 1, 1, 1, 1, 1, 1, 1, 1, 5, 5, 1, 2, 2, 1, 3, 7, 2,
       4, 1, 1, 1, 1, 5, 1, 2, 2, 5, 4, 6, 4, 2, 2, 2, 4, 2, 6, 4, 6, 2,
       2, 0, 1, 0, 3, 6, 0, 0, 0, 0, 2, 2, 0, 7, 2, 3, 0, 0, 2, 0, 2, 2,
       2, 1, 2, 1, 4, 2, 2, 2, 2, 2, 3, 2, 5, 4, 5, 3, 1, 3, 0, 7, 0, 3,
       3, 3, 3, 4, 2, 3, 5, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 0, 0, 3, 3, 3,
       5, 4, 2, 3, 3, 4, 2, 6, 3, 0, 2, 3, 3, 6, 3, 3, 4, 6, 0, 4, 1, 0,
       0, 3, 2, 4, 5, 5, 4, 3, 4, 3, 4, 0, 6, 6, 6, 6, 1, 4, 2, 4, 6, 0,
       4, 6, 4, 2, 4, 4, 6, 4, 5, 2, 4, 2, 4, 4, 5, 2, 6, 6, 5, 3, 5, 1,
       1, 5, 3, 1, 1, 5, 3, 3, 3, 5, 3, 5, 2, 5, 1, 5, 1, 3, 5, 5, 4, 1,
       1, 5, 5, 1, 4, 2, 5, 5, 6, 3, 5, 5, 5, 6, 1, 5, 5, 5, 5, 5, 5, 5,
       4, 5, 6, 6, 6, 7, 6, 6, 4, 6, 4, 4, 6, 6, 6, 7, 0, 6, 6, 6, 3, 6,
       4, 2, 5, 5, 6, 4, 6, 4, 4, 4, 6, 0, 4, 4, 4, 5, 2, 5, 6, 0, 6, 4,
       3, 4, 2, 6, 6, 6, 7, 5, 1, 1, 7, 1, 1, 1, 1, 1, 7, 1, 2, 4, 5, 5,
       7, 1, 5, 1, 7, 4, 4, 1])
```

In [6]:
```
y_test = []
for i in range(len(y_pred)):
    if i<(len(angry)):
        y_test.append(0)
    elif i<(len(angry)+len(calm)):
        y_test.append(1)
    elif i<(len(angry)+len(calm)+len(disgust)):
        y_test.append(2)
    elif i<(len(angry)+len(calm)+len(disgust)+len(fearful)):
        y_test.append(3)
    elif i<(len(angry)+len(calm)+len(disgust)+len(fearful)+len(happy)):
        y_test.append(4)
    elif i<(len(angry)+len(calm)+len(disgust)+len(fearful)+len(happy)+len(sad
)):
        y_test.append(5)
    elif i<(len(angry)+len(calm)+len(disgust)+len(fearful)+len(happy)+len(sad)
+len(surprised)):
        y_test.append(6)
    else:
        y_test.append(7)
y_test
```

```
Out[6]: [0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         0,
         1,
         1,
         1,
         1,
         1,
         1,
         1,
         1,
         1,
```

```
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
```

```
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
```

```
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
5,
```

```
5,
5,
5,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
6,
7,
7,
7,
7,
7,
7,
```

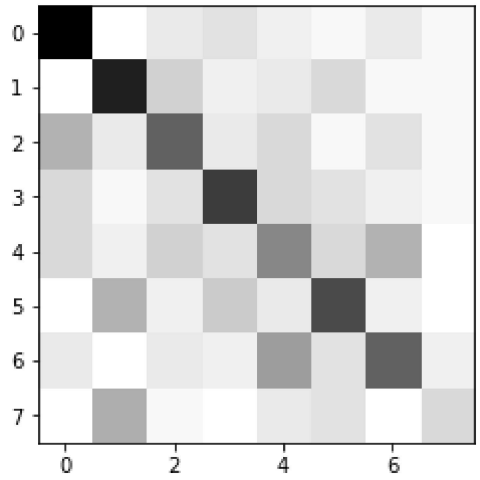
```
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7]
```

In [7]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
cm
```

Out[7]:
```
array([[34,  0,  3,  4,  2,  1,  3,  1],
       [ 0, 30,  6,  2,  3,  5,  1,  1],
       [10,  3, 21,  3,  5,  1,  4,  1],
       [ 5,  1,  4, 26,  5,  4,  2,  1],
       [ 5,  2,  6,  4, 16,  5, 10,  0],
       [ 0, 10,  2,  7,  3, 24,  2,  0],
       [ 3,  0,  3,  2, 13,  4, 21,  2],
       [ 0, 11,  1,  0,  3,  4,  0,  5]])
```

In [8]:
```
import matplotlib.pyplot as plt
plt.imshow(cm, cmap='binary')
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x7f36857daa90>
```



In [9]:
```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_
score
Accuracy_Score = accuracy_score(y_test,y_pred)
Precision_Score = precision_score(y_test, y_pred, average="macro")
Recall_Score = recall_score(y_test, y_pred, average="macro")
F1_Score = f1_score(y_test, y_pred, average="macro")
```

In [10]:
```
Accuracy_Score
```

Out[10]: 0.49166666666666664

In [11]:
```
Precision_Score
```

Out[11]: 0.4854891213637089

In [12]:
```
Recall_Score
```

Out[12]: 0.4739583333333333

In [13]:
```
F1_Score
```

Out[13]: 0.4726632694812786