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, "svm", "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",
"svm"))
for i in calm:
    c = np.append(c,aT.fileClassification("total/test/calm/"+i, "svmSMtemp","s
vm"))
for i in disgust:
    c = np.append(c,aT.fileClassification("total/test/disgust/"+i,"svmSMtemp",
"svm"))
for i in fearful:
    c = np.append(c,aT.fileClassification("total/test/fearful/"+i,"svmSMtemp",
"svm"))
for i in happy:
    c = np.append(c,aT.fileClassification("total/test/happy/"+i,"svmSMtemp","s
vm"))
for i in sad:
    c = np.append(c,aT.fileClassification("total/test/sad/"+i,"svmSMtemp","sv
m"))
for i in surprised:
    c = np.append(c,aT.fileClassification("total/test/surprised/"+i,"svmSMtem
p","svm"))
for i in neutral:
    c = np.append(c,aT.fileClassification("total/test/neutral/"+i,"svmSMtemp",
"svm"))
c = np.reshape(c,(-1,8))
```

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

Out[4]:
```
array([[0.33237965, 0.01484192, 0.12970159, ..., 0.09214423, 0.06913406,
        0.0218061 ],
       [0.45820567, 0.00507741, 0.15353102, ..., 0.04949885, 0.10658104,
        0.0268332 ],
       [0.08131753, 0.00581583, 0.02342121, ..., 0.03015495, 0.09955517,
        0.00459473],
       ...,
       [0.02541572, 0.15218518, 0.03570282, ..., 0.16251699, 0.12432099,
        0.24054508],
       [0.07524657, 0.0665574 , 0.08680857, ..., 0.09764536, 0.23650363,
        0.23366782],
       [0.0397886 , 0.21760193, 0.06175633, ..., 0.15856276, 0.09937567,
        0.20853225]])
```

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

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

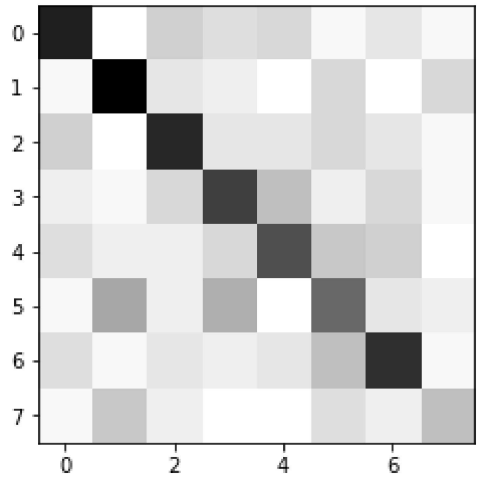
```
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7]
```

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

Out[7]:
```
array([[28,  0,  6,  4,  5,  1,  3,  1],
       [ 1, 32,  3,  2,  0,  5,  0,  5],
       [ 6,  0, 27,  3,  3,  5,  3,  1],
       [ 2,  1,  5, 24,  8,  2,  5,  1],
       [ 4,  2,  2,  5, 22,  7,  6,  0],
       [ 1, 11,  2, 10,  0, 19,  3,  2],
       [ 4,  1,  3,  2,  3,  8, 26,  1],
       [ 1,  7,  2,  0,  0,  4,  2,  8]])
```

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

Out[8]:
```
<matplotlib.image.AxesImage at 0x7fc6aeb05b10>
```



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]: 0.5166666666666667

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

Out[11]: 0.5100238696438465

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

Out[12]: 0.5052083333333333

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

Out[13]: 0.5062151359207587