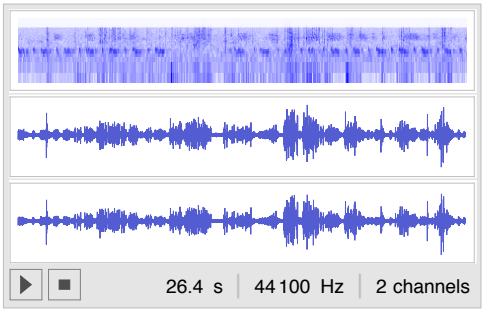
In[1]:=
```
SetDirectory[NotebookDirectory[]];
```

In[2]:=
```
ReduceSamplingRate[signal_,k_]:=Module[{tmp},
tmp=signal;
tmp[[1,1,1]]=tmp[[1,1,1]][[;;;;k]];
tmp[[1,1,2]]=tmp[[1,1,2]][[;;;;k]];
tmp[[1,2]]=tmp[[1,2]]/k;
Return[tmp];]
```

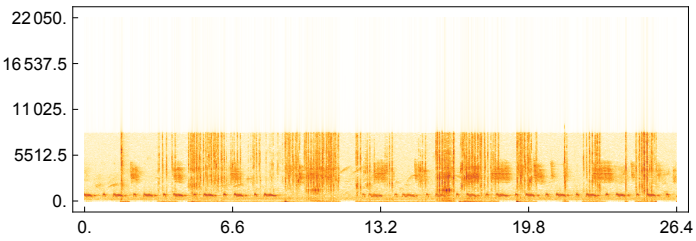
In[3]:=
```
snake = Import["snake.wav", "wav"]
```

Out[3]=



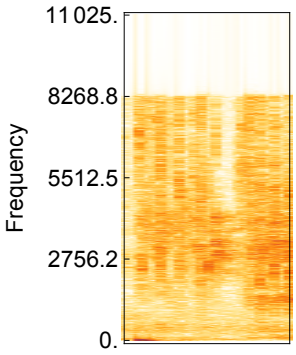
In[113]:=
```
Spectrogram[snake]
```

Out[113]=



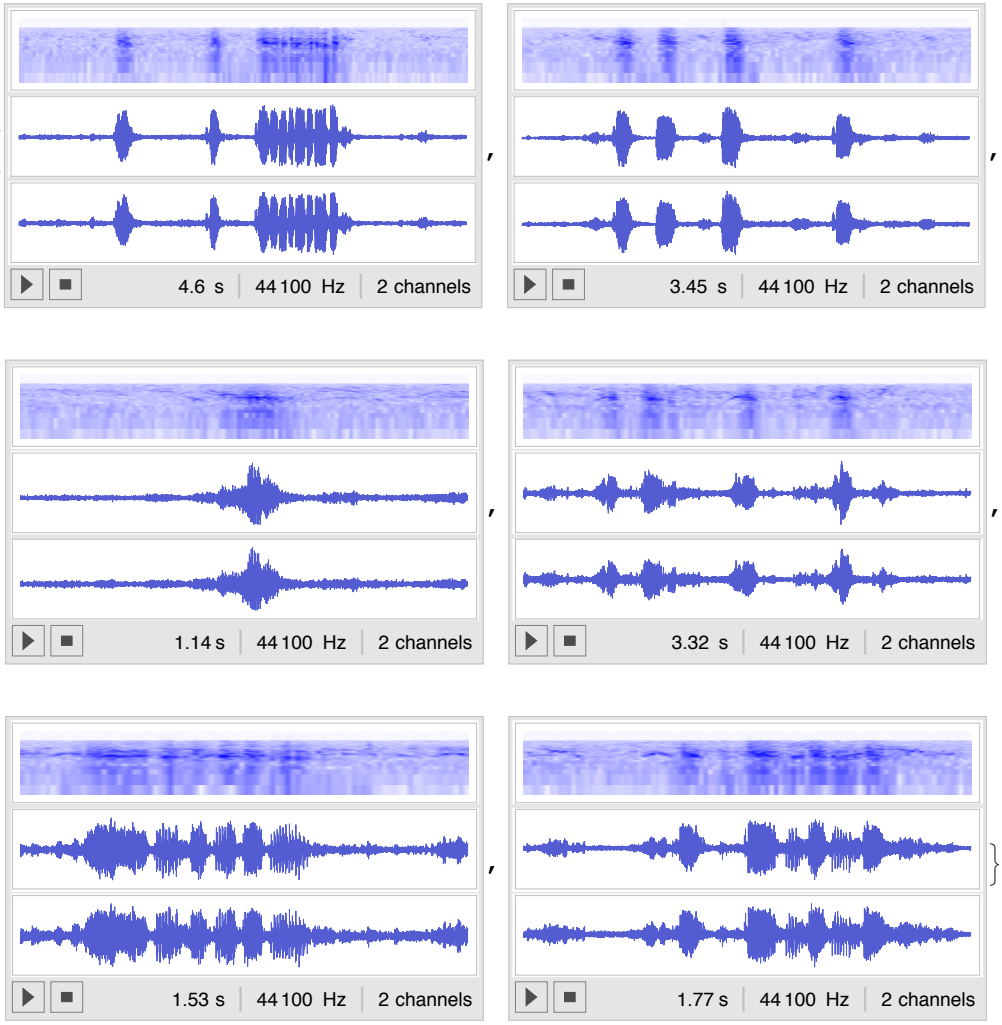
In[114]:=
```
Spectrogram[ReduceSamplingRate[snake, 2],
 PlotRange → {{9, 10}, {0, 11000}}, AspectRatio → 2,
 FrameLabel → {"Frequency", None}, LabelStyle → Medium, ImageSize → 150]
```

Out[114]=



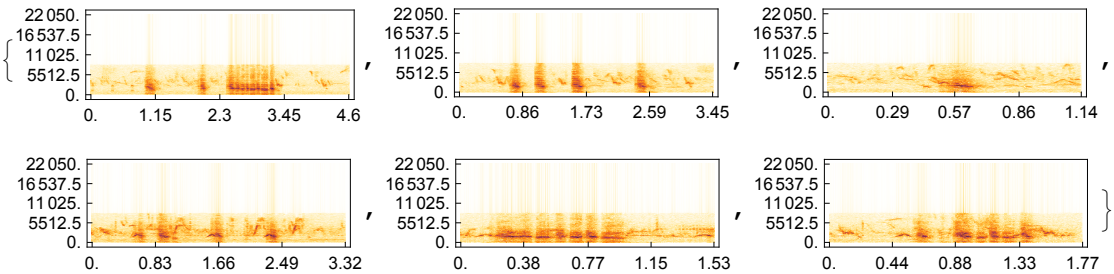
In[4]:=
```
eagles = Table[Import["eagle_" <> ToString[i] <> ".wav", "wav"], {i, 1, 6}]
```

Out[4]=



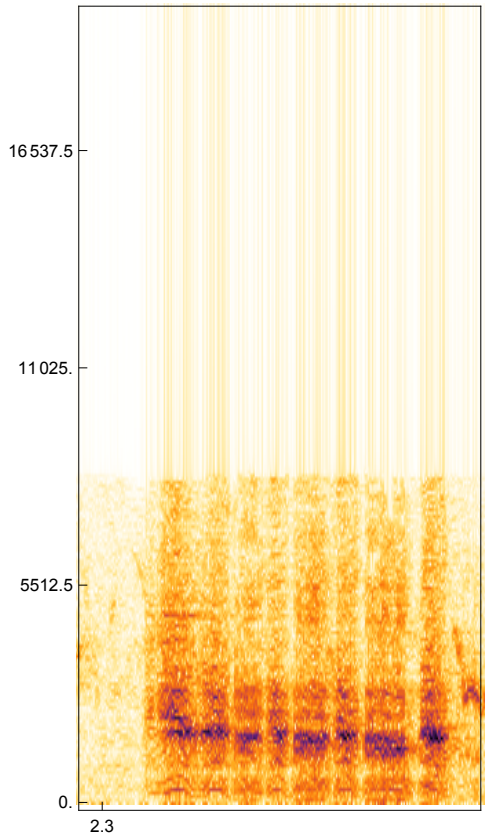
In[115]:=
```
Spectrogram /@ eagles
```

Out[115]=



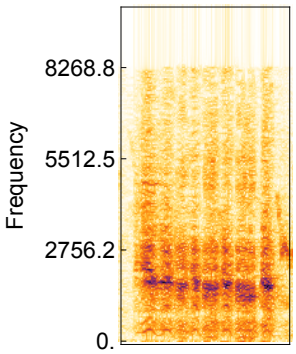
In[116]:=
```
Spectrogram[eagles[[1]], 512,
 PlotRange → {{2.3, 3.3}, {0, 20 000}}, AspectRatio → 2]
```

Out[116]=



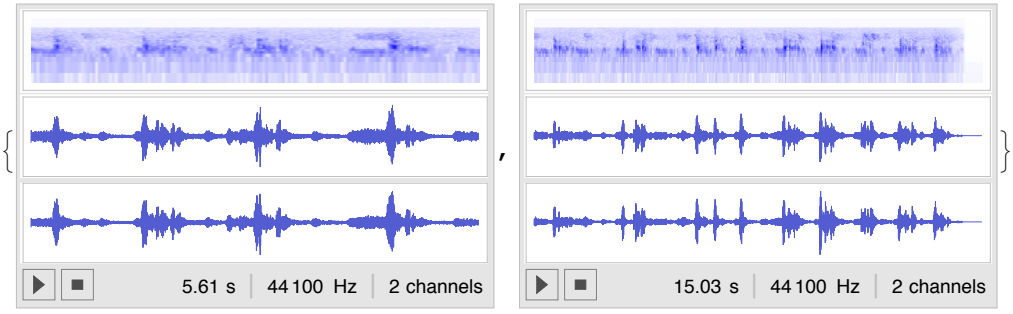
In[117]:=
```
Spectrogram[ReduceSamplingRate[eagles[[1]], 2],
 PlotRange → {{2.4, 3.3}, {0, 10 000}}, AspectRatio → 2,
 FrameLabel → {"Frequency", None}, LabelStyle → Medium, ImageSize → 150]
```

Out[117]=



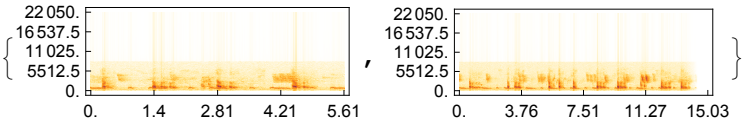
In[5]:=
```
leopards = Table[Import["leopard_" <> ToString[i] <> ".wav", "wav"], {i, 1, 2}]
```

Out[5]=



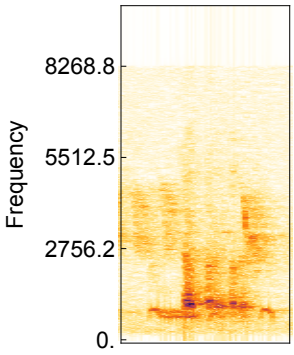
In[118]:=
```
Spectrogram /@ leopards
```

Out[118]=



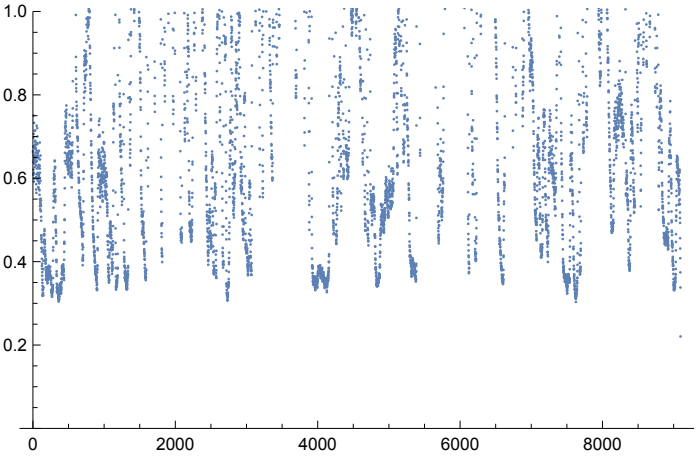
In[119]:=
```
Spectrogram[ReduceSamplingRate[leopards[[2]], 2],
 PlotRange → {{8, 9}, {0, 10 000}}, AspectRatio → 2,
 FrameLabel → {"Frequency", None}, LabelStyle → Medium, ImageSize → 150]
```

Out[119]=



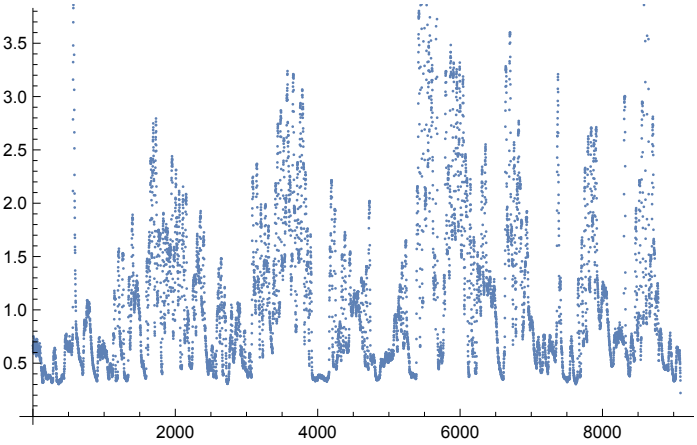
In[120]:=
```
ListPlot[
 Map[Mean, Abs[SpectrogramArray[ReduceSamplingRate[snake, 2], 1024, 64]]],
 PlotRange → {0, 1}]
```

Out[120]=



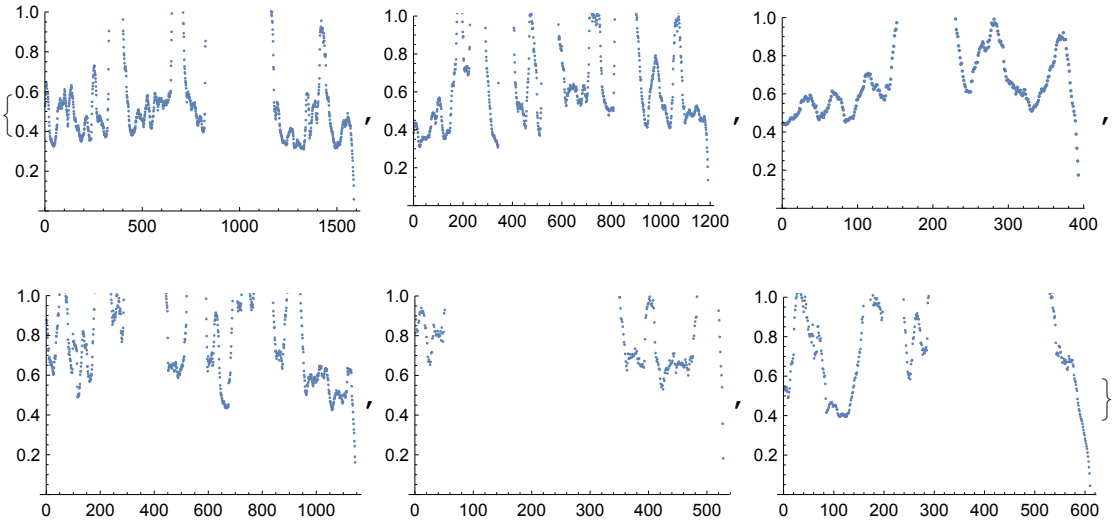
In[121]:=
```
ListPlot[
 Map[Mean, Abs[SpectrogramArray[ReduceSamplingRate[snake, 2], 1024, 64]]]]
```

Out[121]=



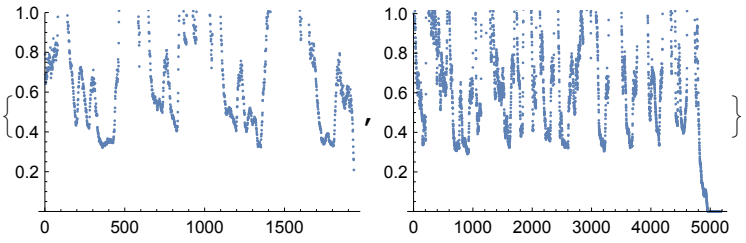
In[122]:=
```
ListPlot[Map[Mean, Abs[SpectrogramArray[ReduceSamplingRate[#, 2], 1024, 64]]],
  PlotRange → {0, 1}] & /@ eagles
```

Out[122]=



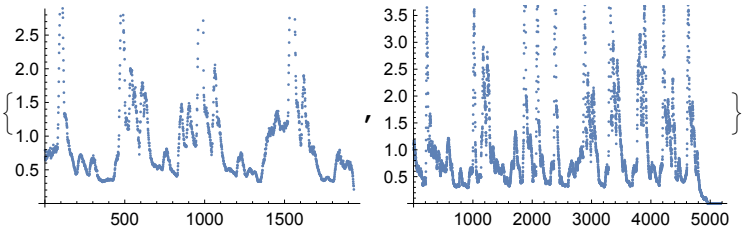
In[123]:=
```
ListPlot[Map[Mean, Abs[SpectrogramArray[ReduceSamplingRate[#, 2], 1024, 64]]],
  PlotRange → {0, 1}] & /@ leopards
```

Out[123]=



In[124]:=
```
ListPlot[Map[Mean,
    Abs[SpectrogramArray[ReduceSamplingRate[#, 2], 1024, 64]]] & /@ leopards
```

Out[124]=



In[6]:=
```
GenFrames[audio_,thresh_]:=Select[Abs[SpectrogramArray[audio,1024,64,HammingWindow]],M
```

Filtering

In[7]:=
```
thresh = 1;
snakeFeats = GenFrames[ReduceSamplingRate[snake, 2], thresh];
eaglesFeats =
  Flatten[GenFrames[ReduceSamplingRate[#, 2], thresh] & /@ eagles, 1];
leopFeats = Flatten[GenFrames[ReduceSamplingRate[#, 2], thresh] & /@ leopards, 1];
Dimensions /@ {snakeFeats, eaglesFeats, leopFeats}
```

Out[11]=
```
{{1603, 1024}, {1307, 1024}, {992, 1024}}
```

In[12]:=
```
rawFeat = Flatten[{snakeFeats, eaglesFeats, leopFeats}, 1];
```

PCA with 99% variance conserved

In[13]:=
```
pca = PrincipalComponents[rawFeat, Method → "Correlation"];
```

In[14]:=
```
varPCA = Variance[pca];
```

In[15]:=
```
tmp = 0.99 Total[varPCA];
For[i = 1, i ≤ Length[varPCA], i++, tmp = tmp - varPCA[[i]];
  If[tmp < 0, Print[i];
   Break[]]];
```

```
274
```

In[17]:=
```
feats = rawFeat[[All, ;; i]];
```

Results

Hierachical clustring

In[18]:=
```
cluster3 =
  FindClusters[feats -> Flatten[{ConstantArray["snake", Length[snakeFeats]],
      ConstantArray["eagle", Length[eaglesFeats]], ConstantArray[
       "leopard", Length[leopFeats]]}, 1], 3, Method → "Agglomerate"];
```

In[19]:=
```
Count[#, "snake"] & /@ cluster3
Count[#, "eagle"] & /@ cluster3
Count[#, "leopard"] & /@ cluster3
```

Out[19]= {1603, 0, 0}

Out[20]= {1307, 0, 0}

Out[21]= {981, 10, 1}

kmeans

In[103]:=
```
kmeans = ClusteringComponents[feats, 3, 1, Method → "KMeans", "RandomSeed" → 1];
kmeans = {kmeans[[ ;; Length[snakeFeats]]],
   kmeans[[Length[snakeFeats] + 1 ;; Length[snakeFeats] + Length[eaglesFeats]]],
   kmeans[[Length[snakeFeats] + Length[eaglesFeats] + 1 ;;]]};
Count[#, 1] & /@ kmeans
Count[#, 2] & /@ kmeans
Count[#, 3] & /@ kmeans
```

Out[105]= {1593, 530, 451}

Out[106]= {10, 777, 13}

Out[107]= {0, 0, 528}

PAM

In[108]:=
```
pam = ClusteringComponents[feats, 3, 1, Method → "PAM", "RandomSeed" → 1];
pam = {pam[[ ;; Length[snakeFeats]]],
    pam[[Length[snakeFeats] + 1 ;; Length[snakeFeats] + Length[eaglesFeats]]],
    pam[[Length[snakeFeats] + Length[eaglesFeats] + 1 ;;]]};
Count[#, 1] & /@ pam
Count[#, 2] & /@ pam
Count[#, 3] & /@ pam
```

Out[110]= {1550, 349, 338}

Out[111]= {53, 958, 49}

Out[112]= {0, 0, 605}