Chapter 3

Content Analysis for Retrieval

All lands and towns are learner's own
Why not till death learning goes on!
- Thirukkural (397)

3.1 INTRODUCTION

Recognizing sources in the environment from the sounds they produce is one of the primary functions of the auditory system. Sound producing objects have acoustic properties, which are the result of the production process. These properties enable us to recognize sound sources by listening. The property includes the type of excitation, the physical construction, and the shape and size of the resonance structures. The type of excitation varies from one to another, and has significant influence on the sound.

The goal of automatic auditory scene analysis is to create computer systems that can learn to recognize the sound sources in a complex auditory environment. In our proposed system, the vocalization of six different wild animals was considered. This includes sounds produced by bear, lion, gorilla, elephant, wolf and eagle. The data of the selected six wild animals is summarized in Table 3.1. The table contains English names and family, class, order and vocalizations (sound types).
This chapter is organized as follows, Section 3.2 gives a survey of audio sound source recognition related works, Section 3.3 explains about the sound segmentation procedure, Section 3.4 briefs about the feature extraction, Section 3.5 explains about the feature selection and Section 3.6 briefs the neural network classifiers and Section 3.7 describes comparative results and analysis. Section 3.8 concludes this chapter.

3.2 AUDIO SOUND SOURCE RECOGNITION – SURVEY

Audio data may be coarsely divided into three classes: speech, music, and environmental sounds. Wild animal sounds are a domain of environmental sounds that has not been investigated yet in detail. Some investigations consider animal sounds among other classes of sound (Kim et al. 2004; Guo et al. 2003). A survey of audio sound source recognition related work is performed in this Section.

3.2.1 Sound Segmentation

Segmentation is an important preprocessing step of audio analysis. It is employed to discriminate different types of sound such as speech, music, environmental sounds and combinations of these. Schreirer et al. (1997) separated music and speech with low level features. They apply Spectral Centroid, Spectral Flux (SF), Zero Crossing Rate (ZCR), Spectral Roll-off, and Percentage of Low Energy Frames to represent the audio signal. Different classification techniques such as Gaussian Mixture Model (GMM) and Nearest Neighbor (NN) are used to separate speech from music based on these features. Based on successful segmentation of an audio stream, different audio types can be further analyzed. The most intensive research took place in the area of speech recognition.
3.2.2 Recognition of environmental sounds

Environmental sound recognition concerns the identification of sounds that do not originate from speech or music. The range of environmental sounds is extremely wide. Hence, most investigations concentrate on a restricted domain. A popular research field is audio recognition in broadcasted video. Liu et al. (1997) recognized the scene content of TV programs (e.g. weather reports, advertisement, basketball and football games) by analyzing the audio track of the video. They extract Pitch, Volume Distribution, Frequency Centroid and Bandwidth to characterize TV programs. Classification is performed by a separate neural network for each class.

There are different groups of audio retrieval techniques. Numerical representation of signals by features, is common to all methods. Approaches can be grouped by the way similarity among different signals is detected. A straightforward technique is to apply a distance measure directly to the features. Wold et al. (1996) developed a content-based audio retrieval system (named as Muscle Fish) that distinguishes classes such as animals, machines, musical instruments, telephone, etc. They extract features such as loudness, pitch, brightness and bandwidth. Similarity is measured using a weighted Euclidean distance (Mahalanobis, 1936). Classification is accomplished by the nearest neighbor rule. An alternative to directly measuring similarity is the use of artificial intelligence techniques such as Support Vector Machines (SVM) (Cortes et al. 1995), Hidden Markov Models (HMM) or Artificial Neural Networks (ANN).
An early example in the domain of audio processing is represented by Feiten et al. (1994). The authors apply a self-organizing neural network to cluster similar sounds. Another way of classification is based on template matching (Foote et al. 1997). The author extracts MFCC features from the audio signal and clusters the feature space into distinct cells with a quantization tree (Q tree). Histograms are considered as templates. They represent the distribution of feature vectors over the partitions of the tree. Templates are compared by distance measures (e.g. Euclidean distance or cosine distance).

A challenging area of environmental sound recognition is life logging. This research field is concerned with continuously analyzing the environmental sounds of a human user. From this information a diary is built where major events and the user's activities are stored. Fundamental research in the domain of life logging is performed in the "Forget-me-not" system presented by Lamming et al. (1994). "Forget-me-not" is a mobile application that analyzes the activities of a user in his office. This includes monitoring the workstation, telephone, printer and the location of the user.

Another area of interest is surveillance and intruder detection. A broad survey of audio features and classification techniques, in context of automatic surveillance is given by Cowling et al. (2004). Zhang et al. (1999) proposed a multilevel classification. First they apply a coarse level segmentation to separate speech, music and environmental sound. In a second step an HMM is considered to analyze environmental sounds (e.g. footstep, laughter, rain, windstorm).
3.2.3 Recognition of Animal sounds

Only few studies have been done on automatic recognition of animal vocalization and efficient parameterization of animal sounds. In (Anderson et al. 1996; Kogan et al. 1998), dynamic time warping and hidden Markov models were used for automatic recognition of songs of birds like Zebra Finches and Indigo Puntings. In these studies syllables were represented by spectrograms. Comparison of spectrograms is computationally demanding and, in the case of field recordings, they often include also environmental information that is not relevant for recognition of bird species.

In (McIlraith et al. 1997), were tested recognition of songs of six birds common in Manitoba, Canada. In this work songs were represented with spectral and temporal parameters of the song. Dimensionality of the feature space was reduced by selecting features for classification by means of their discriminative ability. Neural networks were used for classification of the songs. Training of the neural networks is computationally demanding, but classification with the network is relatively fast. Also back-propagation algorithm weights automatically different features.

Nelson (1989) conducted a study of discriminative ability of different features in sounds of birds like Field Sparrow and Chipping Sparrow against 11 other bird species. They noted that features have different classification ability in context of different species. They use canonical discriminant analysis to determine and select features that maximize the recognition result.

Mitrovic et al. (2006) tried to identify an efficient method for automatically distinguishing between sounds of different animals. They have chosen four animals, namely birds, cats, cows, and dogs. Sounds by birds and cats respectively by cows and dogs show significant similarity on a perceptual level.
The experiments were conducted with three different classifiers SVM, NN and LVQ classifiers.

Kim et al. (2004) presented an audio indexing system using MPEG-7 features. They apply Audio Spectrum Basis (ASB) and Audio Spectrum Projection (ASP) descriptors to distinguish classes such as "Dog", "Bell", "Water", and "Baby" with HMMs. They show that MPEG-7 descriptors perform similar to MFCC. SVMs are successfully applied to environmental sound recognition in (Guo et al. 2003). The authors compare and combine cepstral features (MFCCs) with perceptual features (Total Spectrum Power, Subband Powers, Brightness, Bandwidth, and Pitch). In (Guo et al. 2003) perceptual features outperform cepstral features. Best results are reached by a combination of both. Guo et al. (2003) also shown that SVM performs better than NN and KNN.

3.2.4 Speaker Recognition

Another field of research is classification of the speaker (e.g. for customization issues or authentication) (Reynolds et al. 1995). In the area of multimodal dialog systems recognition of human emotions from audio gains focus (Chiu et al. 1994). The different areas of speech processing are a source to survey state-of-the-art audio features. Beside speech recognition music information retrieval (MIR) gained importance through the availability of huge amounts of digital music. MIR consists of classification and structural analysis. Classification concerns recognition of instruments, artists and genres. Multiple speech recognition features are applicable to the classification of music. Beside classical recognition of speech Rabiner et al. (1993) focus on recognition of the spoken language.
Aizawa et al. (2005) presents a life logging system that captures video and audio. Audio information is considered to detect human voice to recognize conversation scenes. The system supports GPS and provides inertial trackers to measure motion. Additionally it has access to documents, web pages, and emails. Applications discussed in this Section prove the importance of environmental sound recognition for future information systems.

A well investigated problem is highlight detection in sport videos. Tjondronegoro et al. (2004) retrieved crucial scenes in soccer games by analyzing playbreaks. Whistles, that often refer to play-breaks in sports, are detected using Spectral Energy within an appropriate frequency band. Another indicator for highlights is the audience. Excitement is quantified by Loudness, Silence and Pitch. A similar approach is followed by Xu et al. (2004). They analyze keywords in commentator speech and audience which are relevant to important actions of the game. They apply an HMM trained with low level features (Energy and MFCCs including delta and double delta features) to recognize the keywords. Investigations presented by Rui et al. (2000) address extraction of highlights in baseball games. Besides, visual features the authors extract audio features (e.g. MFCC, Pitch, and Entropy). An SVM detects excitement of the audience. Template matching is applied for baseball hit detection. These two audio cues are combined to improve quality of highlight detection.

### 3.2.5 Music Instrument Recognition

Liu et al. (2001) distinguished between instruments (e.g. Brass, Keyboard, and String) by extracting features such as ZCR, Short Time Energy (STE), Bandwidth, Pitch, Formant Frequencies and Mel-Frequency Cepstral Coefficients (MFCC).
These features are computed from short frames of the audio signal. The mean and standard deviations of the features over all frames add up to the final feature vector that represents the signal. Classification is performed by GMM and NN. Music genre classification is addressed by Grimaldi et al. (2003). In this paper the authors propose the discrete wavelet packet decomposition transform to distinguish music genres. Structural music analysis tries to extract similarities and recurrences in a piece of music. A comprehensive structural analysis is performed in (Maddage et al. 2004). Autocorrelation is computed to extract Rhythm from the wavelet-decomposed signal. Pitch Class Profiles in combination with HMM separate chords. Vocal and instrumental sections are characterized in terms of Octave Scaled Cepstral Coefficients (OSCC). An SVM trained with OSCC features separates vocal from instrumental sections.

3.3 SOUND SEGMENTATION

The different elements composing the system will be described in the following order: the sound segmentation, the feature extraction functions module, feature selection module, its link to Neural Networks block. A Sound Segmentation process first ensures the suppression of silent portions of the animal sound signals. When the signal drops under a certain level, the portion is discarded, because features extracted from such part present completely incorrect information to the classifier.

Fractal dimension is widely used for image segmentation. Petros et al. (1999) attempted speech recognition using fractal based segmentation. In this research we have used fractal dimension - D to segment the animal vocalizations. Fractal
dimension of the wild animal sound signal has been computed using one of the most popular methods, 1-D Box-counting (Minkowski dimension) method.

The sampling rate of the sound data, Fs, was 44.1 kHz and 16-bit accuracy was used. The system was developed in the Matlab environment, and the Signal Processing Toolbox was utilized. The input animal vocalization signal was divided into 1024 samples size frames and then normalized. The fractal dimension of the frame has been calculated using 1-D box counting algorithm. The mean of the results acquired with box sizes of 2 to 64 was used as a fractal dimension D for that block. Values computed with box sizes above 64 have no significant information about the signal. Portion of the signal with fractal dimension FD below 1.95 was chosen for feature extraction and the rest was discarded.

<table>
<thead>
<tr>
<th>English name</th>
<th>Family</th>
<th>Class</th>
<th>Order</th>
<th>Vocalizations (sound types)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bear</td>
<td>Ursidae</td>
<td>Mammalia</td>
<td>Carnivora</td>
<td>Moan, Bark, Huff, Growl, Roar</td>
</tr>
<tr>
<td>Eagle</td>
<td>Accipitridae</td>
<td>Aves</td>
<td>Falconiformes</td>
<td>Squeak, chirp</td>
</tr>
<tr>
<td>Asian Elephant</td>
<td>Elephantidae</td>
<td>Mammalia</td>
<td>Proboscidea</td>
<td>Rumbling, trumpet, snort, bark, roar, cry, chirp</td>
</tr>
<tr>
<td>Gorilla</td>
<td>Hominidae</td>
<td>Mammalia</td>
<td>Primates</td>
<td>Hoot, scream, rumble, pig grunt, chest beat, laugh</td>
</tr>
<tr>
<td>Lion</td>
<td>Felidae</td>
<td>Mammalia</td>
<td>Carnivora</td>
<td>snarl, purr, hiss, cough, meow, woof, roar</td>
</tr>
<tr>
<td>Wolf</td>
<td>Canidae</td>
<td>Mammalia</td>
<td>Carnivora</td>
<td>Howl, Growl, bark-howl, whuff, whimper</td>
</tr>
</tbody>
</table>

*Table 3.1 Details of the selected six wild animals*
3.4 FEATURE EXTRACTION

Feature extraction is a process where a segment of an audio is characterized into a compact numerical representation. There are many features that can be used to characterize audio signals. Generally they can be grouped into five categories: temporal, spectral, perceptual, harmonic and statistical. In Section 2.3 we described several audio features that are used in this system.

The feature set of the proposed system contains 30 distinct features (85 features values). The complete list of features is enumerated in Table 2.1. Most of the features for the proposed system have been chosen from MPEG-7 Low-level audio descriptors (2004). Rests of the features are conventional and experimented by researchers proven with good results. There is no publicly available reference database of animal sounds. Animal vocalization sound files for this experiment were obtained from an internet search. All of the digital sound files were then converted into wav files and used for the experiment. The system has been trained with 300 test sets and has been tested with a different set of 300 test signals.

All the features except Temporal centroid and Log-attack time are frame based and are extracted from the segmented frame of 1024 samples. Temporal features are extracted from the time samples directly. Spectral features are computed from Short-Time Fourier Transform (STFT) of the signal. Classification methods are sensitive to the scale of the features, especially in relation to each other. Normalization of the feature set takes care of this. The max of training set was used to select normalization parameters, and these adjustments were applied to all feature sets uniformly.
3.5 FEATURE SELECTION

Feature selection is the process of removing features from the feature set which are less important with respect to the classification task to be performed. Feature selection will also be useful to reduce the processing power required for the classifier and to improve the classification accuracy as well.

3.5.1 Introduction

Feature selection (FS) is usually applied as a pre-processing step in machine learning tasks. FS is employed in different applications with a variety of purposes: to overcome the curse of dimensionality, to remove irrelevant and redundant features thus improving classification performance, to streamline data collection when the measurement cost of attributes are considered, to speed up the classification model construction, and to help unravel and interpret the innate structure of datasets.

It is worth noting that even though some machine learning algorithms perform some degree of feature selection themselves (such as classification trees), feature space reduction can be useful even for these algorithms. Reducing the dimensionality of the data reduces the size of the hypothesis space and thus results in faster execution time.

In general, feature selection techniques can be split into two categories - filter model and wrapper model. The filter model relies on some intrinsic characteristics of data to select features without involving classification learning; the wrapper
model, typically uses a classifier to evaluate feature quality. The wrapper model is often computationally more expensive than the filter model, and its selected features are biased toward the classifier used. In this chapter, we are particularly interested in the filter methods.

### 3.5.2 Feature Selection algorithms

In this Section, we briefly introduce the feature selection algorithms that are classified as Filter Methods according to the computational models they are based on.

#### 3.5.2.1 Laplacian Score

Laplacian Score is to select features that retain sample locality specified by an affinity matrix \( K \). Given \( K \), its corresponding degree matrix \( D \) and Laplacian matrix \( L \), the Laplacian Score of a feature \( f \) is calculated in the following way:

\[
SC_L(f) = \frac{f^\top L \tilde{f}}{f^\top \tilde{f}}, \quad \text{where} \quad \tilde{f} = f - \frac{f^\top D 1}{1^\top D 1}
\]  

.. (3.1)

Using Laplacian Score to select \( k \) features is equivalent to optimizing the following objective:

\[
\min_{i_1, \ldots, i_k} \sum_{j=1}^{k} SC_L(f_{i_j}) = \sum_{j=1}^{k} \frac{\tilde{f}_{i_j}^\top L \tilde{f}_{i_j}}{\tilde{f}_{i_j}^\top D \tilde{f}_{i_j}},
\]

\( i_j \in \{1, \ldots, m\}, \quad p \neq q \rightarrow i_p \neq i_q. \)  

.. (3.2)
In Laplacian Score, features are evaluated independently, therefore the optimization problem defined above can be solved by greedily picking the top k features which have the minimal $SC_L$ values. Since features are evaluated individually, Laplacian Score cannot handle feature redundancy.

### 3.5.2.2 Fisher Score

Given class labels $y = f \{y_1 \ldots y_n\}$, Fisher Score selects features that assign similar values to the samples from the same class and different values to samples from different classes. The evaluation criterion used in Fisher Score can be formulated as:

$$
SC_F(f_i) = \frac{\sum_{j=1}^{c} n_j (\mu_{i,j} - \mu_i)^2}{\sum_{j=1}^{c} n_j \sigma_{i,j}^2}.
$$

(3.3)

Above $\mu_i$ is the mean of the feature $f_i$, $n_j$ is the number of samples in the $j$th class, and $\mu_{ij}$ and $\sigma_{ij}$ are the mean and the variance of $f_i$ on class $j$, respectively. Fisher Score is a special case of Laplacian Score, when $n_i$ is the number of instances in $l$-th class and when $K$ is defined as:

$$
K_{FIS}^{ij} = \begin{cases} 
\frac{1}{m_i}, & y_i = y_j = l \\
0, & \text{otherwise}
\end{cases}.
$$

(3.4)

Fisher Score is an effective supervised feature selection algorithm, which has been widely applied in many real applications. However as the cases of Laplacian Score,
Fisher Score valuates features individually, therefore it cannot handle feature redundancy.

### 3.5.2.3 Chi-square Score

Chi-square is used to assess two types of comparison: tests of goodness of fit and tests of independence. In feature selection it is used as a test of independence to assess whether the class label is independent of a particular feature. Chi-square score for a feature with \( r \) different values and \( C \) classes is defined as

\[
\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{C} \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}},
\]

..(3.5)

where \( n_{ij} \) is the number for samples with the \( i^{th} \) feature value. And

\[
\mu_{ij} = \frac{n_{*j}n_{i*}}{n},
\]

..(3.6)

where \( n_{*j} \) is the number of samples with the \( i^{th} \) value for the particular feature, \( n_{i*} \) is the number of samples in class \( j \) and \( n \) is the number for samples.

### 3.5.2.4 Gini index

Gini index is a measure for quantifying a feature's ability to distinguish between classes. Given \( C \) classes, Gini Index of a feature \( f \) can be calculated as
The Gini Index can take the maximum value of 0.5 for a binary classification. Smaller the Gini Index, more relevant the feature. Gini Index of each feature is calculated independently and the top k features with the smallest Gini index are selected. It does not eliminate redundant features.

### 3.5.2.5 mRMR

Minimum-Redundancy-Maximum-Relevance (mRMR) selects features that are mutually far away from each other, while they still have "high" correlation to the classification variable. mRMR is an approximation to maximizing the dependency between the joint distribution of the selected features and the classification variable.

**Minimize Redundancy**

For discrete variables: $\min W_I, W_I = \frac{1}{|s|^2} \sum_{i,j \in S} I(i,j)$

For continuous variables: $\min W_c, W_c = \frac{1}{|s|^2} \sum_{i,j} |c(i,j)|$ ..(3.8)

**Maximize Relevance**
For Discrete variables: \( \max V_I, V_I = \frac{1}{|S|} \sum_{i \in S} I(h, i) \)

For Continuous variables: \( \max V_F, V_F = \frac{1}{|S|} \sum_{i \in S} F(i, h) \)

where \( S \) is the set of features
\( I(i,j) \) is mutual information between features \( i \) and \( j \)
\( c(i,j) \) is the correlation between features \( i \) and \( j \)
\( h \) = target classes
\( F(i,h) \) is the F-statistic

Maximum Dependency criterion is defined by \( I(S,h) \), that gives the Mutual Information between the selected variables \( S \) and the target variable \( h \).

- For two univariate variables \( x \) and \( y \):
  \[
  I(x; y) = \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dxdy \tag{3.10}
  \]

- For multivariate variables \( S_m \) and the target \( h \):
  \[
  I(S_m; h) = \int \int p(S_m, h) \log \frac{p(S_m, h)}{p(S_m)p(h)} dS_m dh \tag{3.11}
  \]

The minimal Redundancy-Maximal Relevance (mRMR) is one of the fastest feature selection algorithms belonging to the category of filter method. Peng et al. (2005) investigated the significance of the 'minimal Redundancy- Maximal Relevance' (mRMR) feature selection algorithm and compared with many practical feature
selection algorithms. The results showed better classification accuracy using mRMR feature selection. For a detailed comparison of mRMR with other feature selection algorithms are given in (Peng et al. 2005).

In this chapter we have investigated the impact of the addition of mRMR feature selection algorithm towards animal vocalization classification system. For this purpose we evaluated several cases for animal sound classification to select a compact set of features with better classification accuracy. We used an extensive collection of 85 features in the experiments of study. The feature set was later trimmed to 60 features by applying the mRMR feature selection algorithm. Results of both the experiments are presented and analyzed.

3.6 CLASSIFICATION

With classification, classifiers vary in terms of robustness, speed, memory usage and complexity. This Section gives a brief overview of the classification methods and the parameters used. Two different classifiers were used: K-Nearest Neighbor and Multi-Layer Perceptron classifiers.

3.6.1 K-Nearest Neighbor

The K-Nearest Neighbors (KNN) classifier is a typical example of a distance-based classifier. It stores all the training examples and then calculates a distance between the test observation and all the training observations, thus it employs lazy learning by simply storing all training instances. The class of the closest training example is given as the classification result (1- NN), or the class appearing most often among the K nearest training observations (KNN).
A suitable distance metric needs to be chosen with the KNN. We used the Euclidean distance metric in a normalized space that was obtained with the discrete form of the KL-transform. The transform is a special case of the principal component analysis if none of the dimensions is dropped. It is also equal to using the Mahalanobis distance with the same covariance matrix for all classes which is estimated from the whole data, and calculating the distance to all the training examples instead of class means. In the normalized space, the features are uncorrelated and the range of variation of each feature is the same.

The KNN classifier is straightforward to implement, and it can form arbitrarily complex decision boundaries. Therefore it was used in many of our simulations. The problem of the KNN classifier is that it is sensitive to irrelevant features which may dominate the distance metric. In addition, the calculation requires a significant computational load if a large number of training instances is stored. In our proposed system, the KNN classifier with K=11 has been chosen experimentally.

3.6.2 Multi-Layer Perceptron

The back-propagation algorithm has emerged as the workhorse for the design of a special class of layered feed-forward networks known as multilayer perceptrons (MLP). As shown in Figure 3.1, a multilayer perceptron has an input layer of source nodes and an output layer of neurons (i.e., computation nodes); these two layers connect the network to the outside world. In addition to these two layers, the multilayer perceptron usually has one or more layers of hidden neurons, which are so called because these neurons are not directly accessible. The hidden neurons extract important features contained in the input data.
The training of an MLP is usually accomplished by using a back-propagation (BP) algorithm that involves two phases.

- **Forward Phase:** During this phase the free parameters of the network are fixed, and the input signal is propagated through the network of Figure 3.1.
layer by layer. The forward phase finishes with the computation of an error signal

\[ e_i = d_i - y_i \]  

where \( d_i \) is the desired response and \( y_i \) is the actual output produced by the network in response to the input \( x_i \).

- **Backward Phase.** During this second phase, the error signal \( e_i \) is propagated through the network of Figure 3.1 in the backward direction, hence the name of the algorithm. It is during this phase that adjustments are applied to the free parameters of the network so as to minimize the error \( e_i \) in a statistical sense.

\[ \phi(v) \]

\[ \alpha = -1.719 \]

\[ 0 \]

\[ 1.0 \]

\[ -1.0 \]

\[ -\alpha = -1.719 \]

\[ v \]

*Figure 3.2 Antisymmetric activation function.*
In our proposed system the MLP network was trained with back propagation using scaled conjugate gradient optimization. Back-propagation learning may be implemented in one of two basic ways, as summarized here:

1. Sequential mode: (also referred to as the on-line mode or stochastic mode): In this mode of BP learning, adjustments are made to the free parameters of the network on an example-by-example basis. The sequential mode is best suited for pattern classification.

2. Batch mode: In this second mode of BP learning, adjustments are made to the free parameters of the network on an epoch-by-epoch basis, where each epoch consists of the entire set of training examples. The batch mode is best suited for nonlinear regression.
The back-propagation learning algorithm is simple to implement and computationally efficient in that its complexity is linear in the synaptic weights of the network. However, a major limitation of the algorithm is that it does not always converge and can be excruciatingly slow, particularly when we have to deal with a difficult learning task that requires the use of a large network.

We may try to make back-propagation learning perform better by invoking the following list of heuristics:

- Use neurons with antisymmetric activation functions (e.g., hyperbolic tangent function) in preference to nonsymmetric activation functions (e.g., logistic function). Figure 3.2 and Figure 3.3 shows examples of these two forms of activation functions.
- Shuffle the training examples after the presentation of each epoch; an epoch involves the presentation of the entire set of training examples to the network.
- Follow an easy-to-learn example with a difficult one.
- Preprocess the input data so as to remove the mean and decorrelate the data.
- Arrange for the neurons in the different layers to learn at essentially the same rate. This may be attained by assigning a learning rate parameter to neurons in the last layers that is smaller than those at the front end.
- Incorporate prior information into the network design whenever it is available.
One other heuristic that deserves to be mentioned relates to the size of the training set, \( N \), for a pattern classification task. Given a multilayer perceptron with a total number of synaptic weights including bias levels, denoted by \( W \), a rule of thumb for selecting \( N \) is

\[
N = O\left(\frac{W}{\varepsilon}\right)
\]

where \( O \) denotes "the order of" and \( \varepsilon \) denotes the fraction of classification errors permitted on test data.

### 3.6.3 Classifier Fusion

In our proposed system, we have chosen the method operating on classifiers outputs produced by individual classifiers. The classifier fusion has been done at decision level. We employ the Sum-based and Confidence-based integration strategies to combine two classifiers KNN and MLP. A detailed description about the classifier fusion model has been given in Section 2.5.

### 3.7 CLASSIFICATION AND RETRIEVAL: RESULTS

The aim of the study presented in this chapter is to investigate the impact of the addition of feature selection towards wild-animal vocalization classification. For this purpose some experiments were designed and conducted and experiment results are analyzed in this Section. Our classification results are displayed in Table 3.2 and Table 3.3. In each column, the classification correctness of the wild animal vocalizations over different classification algorithm is presented. The rows of the table show how each animal sound is recognized. Column "No FS" is the results
for the original feature vectors with no feature selection algorithm employed. Column "FS" is the results for compact set of features, minimized using mRMR feature selection algorithm. A significant observation that can be made is that the addition of the feature selection step has significantly improved the performance accuracy of the system. An average improvement of almost 5% is obtained.

<table>
<thead>
<tr>
<th>Animal</th>
<th>KNN</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No FS</td>
<td>FS</td>
</tr>
<tr>
<td>Bear</td>
<td>65.7%</td>
<td>68.7%</td>
</tr>
<tr>
<td>Eagle</td>
<td>76.7%</td>
<td>81.0%</td>
</tr>
<tr>
<td>Elephant</td>
<td>45.0%</td>
<td>48.3%</td>
</tr>
<tr>
<td>Gorilla</td>
<td>54.3%</td>
<td>64.3%</td>
</tr>
<tr>
<td>Lion</td>
<td>35.7%</td>
<td>41.7%</td>
</tr>
<tr>
<td>Wolf</td>
<td>82.7%</td>
<td>80.7%</td>
</tr>
<tr>
<td>Overall</td>
<td>60.0%</td>
<td>64.1%</td>
</tr>
</tbody>
</table>

*Table 3.2 Test Results of Individual Classifiers*
Table 3.3 Test Results of classifier fusion (KNN + MLP)

Table 3.3 contains the recognition result of the classifier fusion experiment. The highest accuracy was obtained with KNN+MLP classifier (Sum-based fusion) using the Minimal Redundancy-Maximum Relevance-based Features Selection. Altogether, 70.3% of the test sounds of the six wild animals were recognized correctly while using both mRMR feature selection and Classifier fusion techniques.
3.8 CHAPTER SUMMARY AND CONCLUSION

In this chapter we presented a system for recognition of vocalization of six different wild animals which yield 70.3% recognition rate while using both mRMR feature selection and Classifier fusion techniques. The recordings of wild animal sounds are from many different sources, like from different web sources. Typically recording and environmental conditions differ from one recording to another and information on these conditions is not available. The system needs to be invariant for these conditions. This holds also for the new recordings because environmental conditions can change abruptly even if recording conditions would be the same. Segmentation is crucial for the further steps of classification because in this phase concurrent segments are extracted from raw recordings. In this work fractal based segmentation method has been tested. We used compact set of features, minimized using minimal Redundancy-Maximum Relevance-based (mRMR) feature selection algorithm, which allowed us to perform better. In this work the focus was to study how wild animal vocalizations can be recognized and classified efficiently. Actually, the identification and extraction of features being representative for a distinct animal vocalization generally is a great challenge. The mRMR feature selection was selected for its ability to analyze and decorrelate the feature set. Results from experiments conducted also show that the two factors, feature selection and classifier fusion affects classification accuracy. The long term objective in this research is to develop methodology for a system that is capable to recognize majority of wild animal vocalizations in field conditions. We will study the reasons why specific animal sound produce high recognition errors and how to better differentiate these animal sounds.