ANALYZING 50 KHZ RATS' VOCALIZATIONS USING MACHINE LEARNING APPROACHES

SHAYAN YAZDANI SANGDEH
Master of Science, Sahand University of Technology, 2015

A thesis submitted
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

NEUROSCIENCE

Department of Neuroscience
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

ANALYZING 50 KHZ RATS' VOCALIZATIONS USING MACHINE LEARNING APPROACHES

SHAYAN YAZDANI SANGDEH

Date of Defense: August 12, 2021

Dr. D. Euston Thesis Supervisor	Associate Professor	Ph.D.
Dr. S. Pellis Thesis Examination Committee Member	Professor	Ph.D.
Dr. M. Tata Thesis Examination Committee Member	Professor	Ph.D.
Dr. A. Luczak Chair, Thesis Examination Committee	Professor	Ph.D.

Dedication

To my sister, mom, and dad

Abstract

Recent studies propose different categorization schemes for rats' 50 kHz vocalizations. This study attempted to differentiate these categories based on spectrographic features extracted manually and using convolutional neural networks (CNN). In order to analyze the separability of the categories, we trained different classifiers on the extracted features and the best performance was achieved by a support vector machine (SVM) algorithm using the features derived from a CNN which yielded an accuracy of 63.67%. The results showed that many call categories have a high degree of overlap, suggesting that rats may also have a difficult time discriminating them. Next, we created a dendrogram using D-prime scores (a separability measure) generated from our SVM classifier. This dendrogram suggested a new grouping of calls into 7 different categories that are highly dissimilar. Finally, we trained another SVM model on the 7 new categories and achieved 77.8% accuracy.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. David Euston for giving me an opportunity to be in his lab and for his continuous guidance and support throughout my master's degree. I would also like to thank my committee members Dr. Sergio Pellis and Dr. Matthew Tata for their support and valuable feedback during committee meetings that helped to find the right direction.

I am also thankful to Candace Burke for recording and providing the vocalization data set and other lab mates for all good memories that we had together.

Finally, I would like to thank my family specially my mother for giving me the superpowers in different stages of my life.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

USV	Ultrasonic Vocalization
AMPH	Amphetamine
SVM	Support Vector Machines
KNN	K Nearest Neighbors
R-CNN	Region-based convolutional neural network
CNN	Convolutional Neural Networks
t-SNE	t-Distributed Stochastic Neighbor Embedding
LDA	Linear Discriminant Analysis
LD	Linear Discriminant
PCA	Principal Component Analysis
UPGMA	Unweighted Pair Group Method with Arithmetic Mean
ReLu	Rectified Linear Unit
Tanh	Hyperbolic Tangent

Chapter 1: Introduction

1.1 **Background Information**

Decoding the information transmitted during animal communication is a growing field with many unanswered questions. However, specifying a clear definition for communication itself, has been one the challenges in this field of study. Communication consists of a sender, a receiver and transmission of encoded information between the two as signals (Green and Marler 1979). Otte (Otte 1974) characterized these signals as: "behavioral, physiological or morphological characteristics fashioned or maintained by natural selection". One communication system which has recently become the focus of intense study is rat vocalizations. Gould and Morgan (1941) reported the first evidence that rats could perceive sounds in ultrasonic range and later Anderson (1953) discovered the ability of rats in making ultrasonic vocalizations (USVs). Since rats are social animals, researchers are interested in understanding the communicative role of USVs produced by rats.

Rodents are playing a crucial rule in modern biomedical research, and they have been a great help for scientist in experiments. Due to the limitations of human health-related experiments and the basic physiological and genetic similarities between rats and humans, rodent models have been widely used in health-related studies, especially in mental health research. For instance, rodent models are used to test the safety and efficacy of the drugs that are aimed to be used by humans. Understanding the purpose of USVs could help scientists in neurobiological research to better understand the behavior of their subjects, thus improving research outcomes.

Forty kHz calls that were first observed by Anderson et al. (1953) were then reported to be produced by rat pups (Allin and Banks 1972). These calls are emitted when pups are separated from their dam (Smith 1976). According to this finding, 40 kHz calls play a communicative role

for the pups. Further studies showed that adult rats produce different types of calls in different frequency ranges and 40 kHz calls are specific to rats' pups. There are two main types of adult rats vocalization: 22 kHz and 50 kHz (Brudzynski 2005, Portfors 2007). The frequency range of 22 kHz calls is between 18 and 28 kHz. On the other side, 50 kHz calls are emitted between 30 and 90 kHz (Wöhr and Schwarting 2013). In addition to their occurrence in different frequency ranges, these calls also have different acoustic features. 22 kHz calls are flat based on spectrographic appearance, while 50 kHz calls are frequency modulated except 50 kHz flat calls. Each of these major categories is also related to a specific emotional state (Brudzynski 2009).

There is still debate in the field as to whether rat vocalizations are actually communicative signals. It has been reported that female rats show solicitation behavior when male rat produce 50 kHz USVs vocalizations (Thomas, Howard et al. 1982). Wöhr and Schwarting (2007) observed that when 50 kHz are played rats try to approach the source of the calls. Another example for communicative role of USVs was found in a recent study showing that flat calls in the 20-30 kHz range could serve as a de-escalating signal (Burke, Kisko et al. 2017). On the other hand, Ågmo and Snoeren (2015) showed that there was no change in rats' sexual and social behavior, when one of the pairs is devocalized.

1.2 **22 kHz Vocalizations**

22 kHz calls are related to anxiety and aggression and could have social and non-social functions. They are emitted when rats are involved in aversive interactions (Thomas, Takahashi et al. 1983, Panksepp, Burgdorf et al. 2004). It has been discovered that these calls also serve as a stop sign when rat is being dominated by another rat (Thomas, Takahashi et al. 1983). Male rats also use 22 kHz calls in sexual interactions to signal the end of intercourse (Barfield and Geyer

1972). In addition, males emit these calls when female rat shows resistant to sexual advances or the male is not successful in mounting (Brown 1979). Not all 22 kHz calls are social in nature. For example, 22 kHz calls are also emitted when rats encounter predators (Blanchard, Blanchard et al. 2005) or experience inescapable shock (Kikusui, Nishizawa et al. 2003) even without other rats present. It has been proposed that the emission of these alarm calls are associated with perirhinal cortex and amygdala regions in the brain (Allen, Furtak et al. 2007).

1.3 **50 kHz Vocalizations**

50 kHz calls are linked with positive affective states. It has been shown that 50 kHz call emission rates increase in situations such as anticipation of play or play interactions (Burke, Kisko et al. 2017), anticipation of electrical stimulation to the brain (Burgdorf, Knutson et al. 2000), and delivery of amphetamine (AMPH) (Burgdorf, Knutson et al. 2001). Burke et al (Burke, Kisko et al. 2017) investigated the relationship between 50 kHz calls and behavioral actions using statistical techniques and found that call types which are produced during running and jumping are different from the ones that produced in slower movements. Given the diverse contexts in which positive calls are used, further understanding of the different 50 kHz calls and the contexts in which they are used may provide important insights into the mechanisms of social communication in rats.

50 kHz calls have different shapes based on their profile in time-frequency representation. Therefore, different subcategories have been proposed for 50 kHz calls. In the simplest categorization scheme proposed, 50 kHz calls can be divided into two categories of flat and frequency-modulated (Burgdorf, Kroes et al. 2008, Wöhr, Houx et al. 2008, Ciucci, Ahrens et al. 2009). Brudzynski (Brudzynski 2009) suggested 5 categories by dividing frequency modulated calls into 4 categories of step, trill, step-trill and “other”. A more detailed version of this type of

categorization has been proposed by Wright et al, which divides 50 kHz calls into 14 subcategories (Wright, Gourdon et al. 2010) as follows (see Figure 1.1):

- **Complex:** Syllables that include two or more directional changes in frequency > 3 kHz
- **Upward ramp:** Syllables with monotonic increase in frequency and a mean slope > 0.2 kHz/ms
- **Upward ramp:** Syllables with monotonic decrease in frequency a mean negative slope > 0.2 kHz/ms
- **Flat:** Syllables that are relatively constant
- **Short:** Syllables with time range less than 12 ms
- **Split:** Syllables with jump to a lower frequency at the middle
- **Step up:** Syllables that instantaneously jumps to a higher frequency.
- **Step down:** Syllables that instantaneously jumps to a lower frequency.
- **Multi-step:** Syllables containing two or more instantaneous jumps in frequency
- **Trill:** Syllables that periodically oscillate. The period id relatively 15 ms.
- **Flat-trill combination:** Trills that are combined with a flat component on one or both sides
- **Inverted-U:** Syllables that look like an inverted U
- **Composite:** Syllables except flat/trill combinations that contains different calls.

On the other hand, Takahashi et al. (Takahashi, Kashino et al. 2010) also categorized 50 kHz calls based on the frequency of the highest peak in their power spectrum. In this categorization scheme, calls were divided into two groups: 40 kHz and 60 kHz. They proposed that the group of 40 kHz calls are related to feeding and the group of 60 kHz calls are associated with moving. Burk et al. (Burke, Kisko et al. 2017) attempted to separate flat calls based on duration and frequency and showed that flat calls grouped into three categories based on frequency. According to the previous studies, calls have been grouped into different categories, but a comprehensive investigation of these categories based on data analysis techniques such as machine learning approaches is required to measure discriminability of the calls.

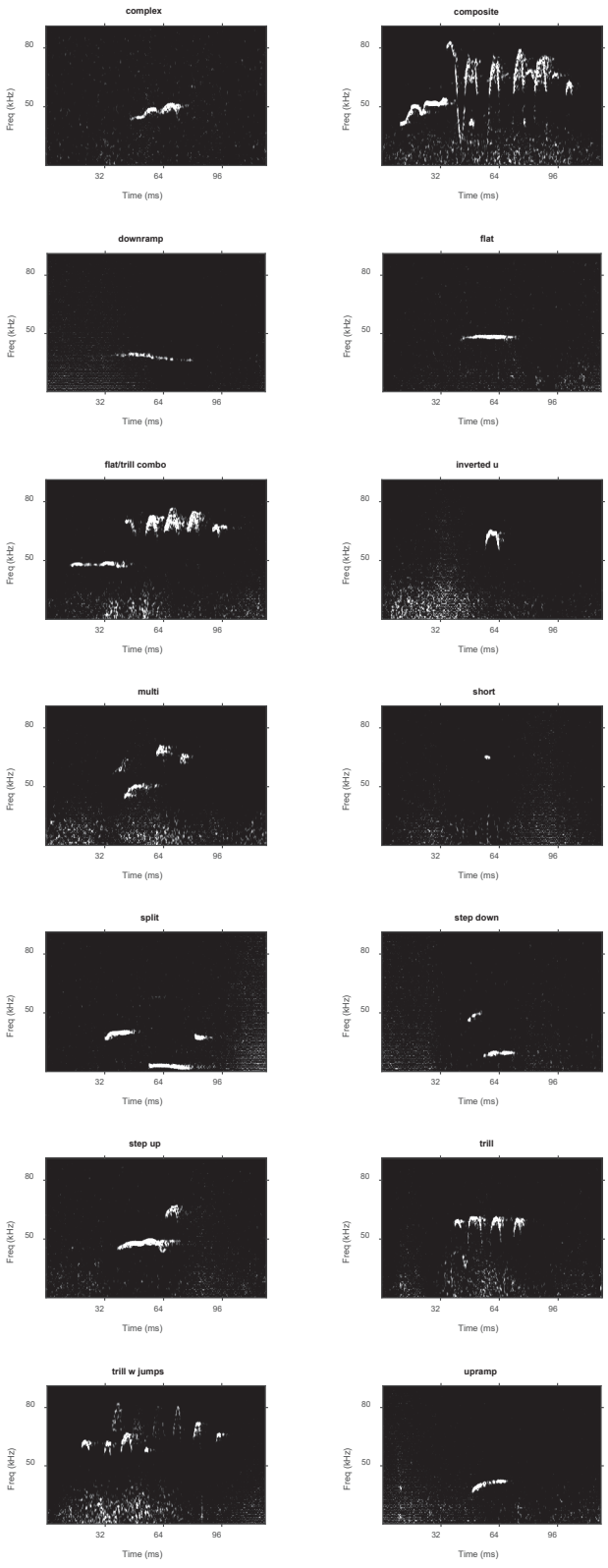

Figure 1.1: 14 types of 50 kHz vocalizations.

1.4 **Machine Learning**

Pattern recognition and machine learning methods are types of new analytical methods that can help us to interpret information in animals' communicative signals. These approaches are a set of methods that can model the relationship between the input data and output data. Therefore, they can help us to find meaningful patterns in data sets without explicitly programming the computer. There are two main types of machine learning approaches: supervised and unsupervised (Goodfellow, Bengio et al. 2016).

In supervised learning, the method tries to find the parameters of a model that can predict the output using the provided labels in the training data set. The data set includes input vectors that are features and respective targets. These input vectors are the values that are measured from the objects or events. Targets are the values that help the algorithm to adjust parameters in such a way that the actual output gets close to the desired output or target that has been defined by human. In other words, the algorithm tries to minimize the error between actual output and desired output. This is an iterative process, and in each iteration, parameters are updated. Iterations are repeated until some pre-determined minimum error is achieved. When the model is completely trained, it is ready to be tested using a new data set to evaluate the accuracy and reliability of the model. In the testing step there is no need to update the parameters and the model can be used for inference. Different steps of the training a model can be summarized as follows:

1. Define the problem
2. Collect and visualize the data
3. Train the model
4. Test the model

5. Collect feedback
6. Refine the model
7. Continue steps 3 to 6 until the model gives the desired output

In general, there are two types of supervised learning algorithms: regression and classification. Regression is used when our prediction is a continuous value. On the other hand, classification methods are used when our prediction is a discrete value or category such as image classification and voice recognition. Recently, classification machine learning methods have attracted many scientists in the field of medical science. For example, many studies have been done on the analysis of electrophysiology data (Liang, Saratchandran et al. 2006, Wang, Nie et al. 2014). In the field of rodent communication, researchers have also utilized machine learning methods to analyze vocalizations both in rats (Coffey, Marx et al. 2019) and mice (Vogel, Tsanas et al. 2019, Fonseca, Santana et al. 2021, Premoli, Baggi et al. 2021). For example, Vogel et al. applied the support vector machine (SVM) method on manually derived acoustic features to replicate human defined categorization in mice vocalizations.

On the other side, in unsupervised learning methods, there is no label to train the model and the algorithm tries to extract some meaningful clusters from the data by itself. This unsupervised categorization of the data is called clustering. However, it might not give the appropriate result since no training is done and no information is given to the method about the target. Therefore, unsupervised methods usually cannot be as accurate as supervised learning method. Table 1.1 shows some examples of methods that are used in unsupervised and supervised machine learning.

Table 1.1: Different machine learning methods

Unsupervised	Supervised	
	Regression	**Classification**
K-Means	Linear Regression	Support Vector Machine
DBSCAN	Logistic Regression	Discriminant analysis
Hierarchical	Ridge Regression	Neural Networks
Hidden Markov Models	Lasso Regression	KNN
	Polynomial Regression	Decision Tree
	Bayesian Linear Regression	Naïve Bayes

1.5 **Available Rat Vocalization Analysis Packages**

Although the purpose of this study was to use machine learning methods to determine the overlap between categories of calls, it is worthwhile to briefly describe available packages for automatic sorting of rat vocalizations. DeepSqueak is a software package by Coffey et al (2019) for analyzing rodent ultrasonic vocalizations. This software can automatically segment or detect vocalizations from spectrograms using an object detection network called Faster Region-based convolutional neural network (R-CNN). It also offers supervised classification and unsupervised clustering of vocalizations using specific methods. For classification, a CNN has been designed, however it has only been trained on eleven categories including: 'Trill', 'Complex-trill combination', 'Complex', 'Down ramp', 'Short', 'Step up', 'Split', 'Flat', 'Up ramp', 'Step down', 'Inverted-U'. The classification performance of DeepSqueak on our data set has been included in

the result section. In the clustering algorithm that Deepsqueak offers, first some manually selected features (e.g., shape, frequency and duration) are extracted and then calls are clustered using the k-means algorithm. In order to extract these features from the calls, a contour is made from the spectrograms. For contour extraction, first, the respective frequency index of maximum amplitude at each time point is detected and then non-tonal features that are related to silence are removed from them. Tonality is measured based on the equation below:

$$1 - \frac{\exp\left(\frac{1}{N}\sum_{n=0}^{N-1}\ln\left(x(n)\right)\right)}{\frac{1}{N}\sum_{n=0}^{N-1}x(n)} \tag{1.1}$$

Where x is the spectrogram and N is the number of points in the spectrogram. In fact, tonality is a measure that shows whether the detected sample is related to the call or background. After constructing the contour from each call, different features are extracted including shape (which is determined using the first derivative of the extracted contour), frequency and duration.

Another package that has been recently proposed is Acoustilytix (Ashley, Snyder et al. 2021). This package offers both call detection and supervised segmentation. For call detection it uses various filters, however details of the method have not been provided. For classification a random forest model has been used. The Model was trained on 5 categories from both 50 kHz and 22 kHz calls. Three of the categories are related to 50 kHz calls including “Fixed frequency calls”, “Frequency modulated” and “Frequency modulated with trills” that are a composite of Wright et al. categories, with two other categories being short and long 22 kHz calls.

1.6 **Summary: Thesis Objectives**

As mentioned before, different categories have been proposed for 50 kHz calls. However, there is still no main categorization scheme that is widely used in the analysis of rat USVs. A proper categorization requires defining appropriate boundaries between the calls which can produce separate categories. This categorization is necessary for conducting further experiments to better understand the communicative role of vocalizations. Therefore, we want to use machine learning methods to find boundaries between calls and compare these to the categories defined by human experts.

This study aims to analyze a large number of calls based on the Wright et al. (2010) categorization scheme. Since one of the categories is composite, i.e., composed of two or more categories, we analyze the calls for the 13 other categories. For each of these 13 call types, we examine the degree to which the call is distinct from all other calls. There are two possibilities. One is that the call forms a discrete category, separate from all other calls. The other possibility is that the cluster overlaps with another cluster. In the former case, we can infer that the two calls are distinct and not part of a continuum. Analyzing all pairs (i.e., one category versus another category) of calls should allow a classification of calls according to spectral similarity, hence providing insights into which categories of calls are likely used by the rat and which are simply a produce of human perceptual grouping. Therefore, first we map our calls in to a high-dimensional "feature space", using some manually selected features, including average frequency, duration and degree of frequency fluctuation as well as features extracted automatically using a CNN. In order to extract features using CNN, vocalizations were converted to spectrograms. The advantage of extracting features from a spectrogram is that it contains both frequency and time information that could be similar to the representation of the activity of the basilar membrane in rats. In the next

step, we try to classify the calls based on these extracted features. Therefore, we use classification machine learning methods such as SVM and K Nearest Neighbors (KNN). These methods classify data by defining decision boundaries between classes. In theory, machine learning provides an upper limit on the accuracy of separating calls. If two calls can be easily separated by an algorithm, they are likely acoustically distinct and therefor good candidates to carry separate communicative signals. If, on the other hand, two calls cannot be separated, it seems unlikely that rats could distinguish them, either, making it highly unlikely that these two calls carry distinct meaning. In practice, any given machine categorization method may be imperfect, so a failure to separate calls could also result from imperfections in the algorithm. Nevertheless, the separability of calls by a sophisticated machine learning algorithm may provide important insights into how calls are actually used by rats.

Chapter 2: Methodology

2.1 **Subjects and Experimental Procedures**

The audio files were picked up from our library of data that had been collected in previous studies (Kisko, Euston et al. 2015, Burke, Kisko et al. 2017). The subjects were long Evans juvenile rats (age 30–40 days). The tests were conducted in a 50 cm × 50 cm × 50 cm clear Plexiglas enclosure which was situated into a soundproof box (inside measurements: 59 cm × 65.5 cm × 81.5 cm). The inside of the enclosure was filled with approximately 1–2 cm of paper-based bedding, in order to reduce the background noise caused by rats' movements. Rats' vocalizations were recorded using a microphone (Model 4939, Brüel & Kjaer, Denmark) with a frequency range of 4-Hz to 100-kHz. The microphone was connected to a Soundconnect™ amplifier (Listen, Inc., Boston, MA, U.S.A.) and digital to analogue conversion obtained using a multifunction processor (model RX6, Tucker-Davis Technologies, Alachua, CA, U.S.A.). The data used in this study were from 29 rats and a total of 69 sessions. The experiment was divided into two trials. First, the subject animal waited for a play partner for 2 minutes (anticipation of play). Second, a familiar partner was introduced, and 10 minutes of play occurred. 24 of the rats were in the anticipation of play procedure. Data were taken from days 1, 2, 3, 4 and 7 with the majority coming from days 1 and 7. 10 of the sessions were from play sessions, where animals interacted for 10 minutes with another animal. In play sessions, vocalizations from both rats are included as it is impossible to assign calls to the other rat. It should be noted that all animals were isolated from their cage partner(s) for 23 hours before each session and in some sessions, rat did not know a partner was going to be introduced. This includes all day 1 sessions and both day 1 and day 7 sessions for the animals in the play control group

2.2 **Vocalization Labeling**

Manually labeling of the vocalizations was performed using a spectrogram-based audio analysis and editing program (Raven Pro software, Cornell, NY). Spectrograms were generated with a 256-sample Hann window. Candace Burke, another graduate student in the Euston lab and I visually categorized spectrograms of 50 kHz vocalizations based on the 14 different categories of 50 kHz vocalizations proposed in (Wright, Gourdon et al. 2010). After labeling, we saved the images of spectrograms in folders related to each category and reviewed the labels. Having the images of spectrograms in folders helped us to compare the calls and revise the labeling. 11411 calls were extracted and labeled in total. Table 2.1 shows the distribution of vocalizations by class.

Table 2.1: Number of calls in each category in the data set.

Category	**Number of Calls**
Trill	3863
Composite	1641
Trill with jumps	1149
Short	990
Complex	798
Upward ramp	662
Step Up	600
Multi	585
Flat	332
Inverted U	272
Split	176
Flat-trill combination	154
Downward ramp	148
Step Down	41

2.3 **Noise Removal**

Since USVs are contaminated by background noise, it is necessary to enhance the quality of the data in order to have more accurate analysis. Therefore, we decided to train a denoising autoencoder to reduce the background noise. Autoencoders are types of artificial neural networks that can be used for data compression, clustering and denoising. An autoencoder consists of two parts: encoder and decoder. The encoder part tries to encode the data into a compressed representation and the decoder part tries to reconstruct the data from the obtained representation. In the training process, the encoder learns how to make a representation from the data set that the decoder can use to reconstruct the data from that representation.

In order to train the denoising autoencoder, 2000 good quality USVs were selected and then the background noise was removed from the spectrogram of the calls. We used “specgram” function in “Matplotlib” library in python programming language and the input parameters were as follows: NFFT=256, noverlap=220, pad_to=1367. In the next step, spectrograms were converted to image arrays by scaling points from 0 to 255. In order to remove the background noise, pixels with values lower than specific threshold were set to zero. This threshold could vary for each spectrogram image. These clean calls were then used as desired outputs in the training process. In order to produce noisy inputs, clean calls were contaminated with background noise manually. The noise was extracted from the parts of the audio that no call occurred. Then, the noise added to the images of spectrograms. Also, we made another set of noisy data with half the level of noise. Therefore, we made 4000 noisy images of the spectrograms in total. Finally, noisy spectrograms were used as inputs and clean spectrograms were used as the targets to train the network.

The autoencoder was implemented in python keras library (https://keras.io/) that was trained in this study is made of convolutional and maxpooling layers. In convolutional layers different filters are used to extract features by sliding them over the input array as shown in figure 2.1. In maxpooling layers, features are downsampled by picking maximum values in local windows. The decoder part of the network consists of two convolutional layers, each followed by a maxpooling layer. The encoder part includes three convolutional layers with one upsampling operation after the first and second convolutional layers. Figure 2.2 shows the complete architecture of the Autoencoder. Rectified Linear Unit (ReLU) and Hyperbolic Tangent (Tanh) were used as activation functions. Note, stride (i.e. the number of pixels that filter shifts) in convolutional layers was equal to (1, 1) and in maxpooling layers was equal to (2, 2).

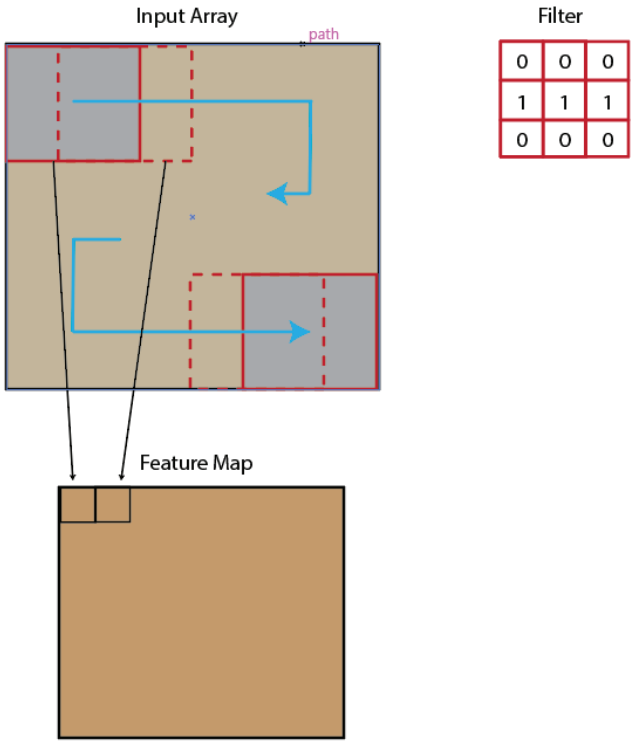

Figure 2.1: Sliding filter over input array (convolution).

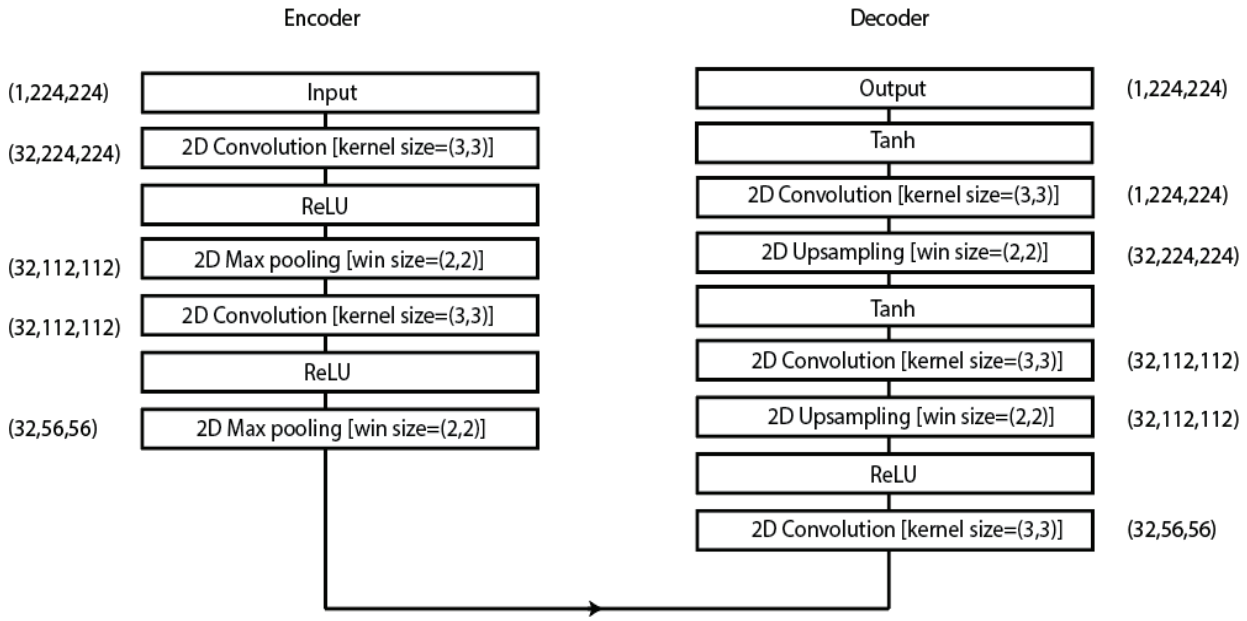

Figure 2.2: Architecture of the autoencoder. The numbers on the sides show the size of each layer

2.4 **Feature Extraction**

For feature extraction, we used a CNN model that is designed for 2D image classification. These networks contain two major parts: feature extractor and classifier. The feature extractor part is comprised of convolutional layers and makes a feature map from the input array using different filters. The classifier part contains fully connected layers and tries to classify data samples using the extracted features in the first part.

One major problem in training deep neural networks is that it requires a large amount of data. Transfer learning is a technique which can be used to overcome this problem. In transfer learning the weights of a network which is trained on task A is used for task B. In other words, the weights of a pretrained network are used to solve a new problem (Shin, Roth et al. 2016). Weights in the layers closest to the output are adjusted to categorize the new inputs, but weights on the

input layers, which do the primitive feature extraction, remain fixed. Since it is hard to provide enough training data for each category of rats' vocalizations, we decided to use a pretrained network called Alexnet feature extraction and transfer learning. Alexnet is a CNN designed for natural image classification (Krizhevsky, Sutskever et al. 2017). The network is trained on the ImageNet data set which contains more than 14 million images of everyday objects including more than 1,000 categories. Its early layers contain detectors for visual primitives very similar to those seen in the mammalian visual system, including orientation-specific edge detectors, which are useful for categorizing a wide range of objects. We reasoned that these same low-level detectors could extract useful features for the categorization of spectrograms, which, like images, are two-dimensional images. As mentioned before, in this study we only use CNN as a feature extractor, and we don't use the classifier part of the network. In order to extract features using the network, the spectrogram of each call is produced and then each point in the spectrogram is normalized between 0 and 255. The "specgram" function in "Matplotlib" library in Python was used to make the spectrograms with following input parameters: NFFT=256, noverlap=220, pad_to=1367. These images are plugged into the pretrained network and 9216 features are extracted from the 5th layer of the network for each call. The Alexnet model and the weights of the network are available in "Pytorch" library (https://pytorch.org/) in Python programming language and we used this package for the feature extraction. In the next step these features will be analyzed using machine learning classifiers.

2.5 **Classification**

In order to analyze the separability of calls we trained different classifiers including SVM and KNN on extracted features as shown in Figure 2.3. We used Python Scikit-learn library

(https://scikit-learn.org/) to train these algorithms. We also trained the classifier part of Alexnet using Python Keras library (https://keras.io/). These methods can help us to examine the possibility of defining boundaries between categories in the feature space. In the rest of this section, these machine learning methods will be briefly described

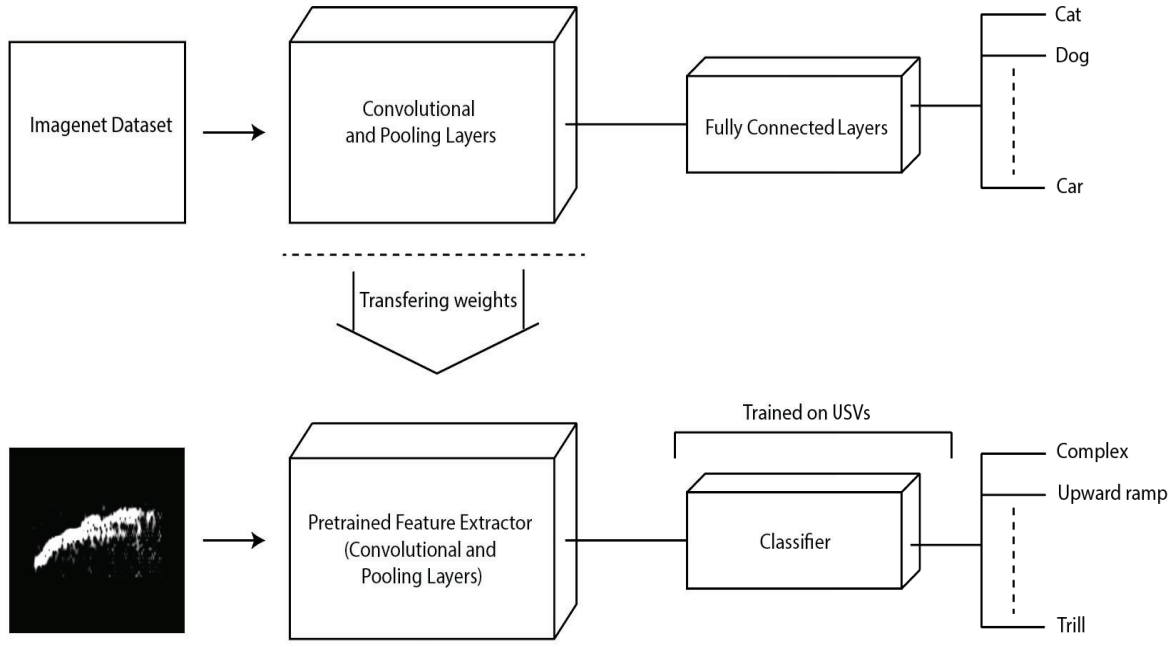

Figure 2.3: Classification architecture that was used in this study.

2.5.1 Support Vector Machine

SVM method is a supervised classifier that tries to find a hyperplane that separates the classes. This hyperplane maximizes the margin between the categories by finding the appropriate support vectors (Cortes and Vapnik 1995, Bishop 2006). Support vectors are a set of points in n-dimensional feature space that define the boundaries of each class. If one these support vectors is changed then the final output will be different. Therefore, finding the best support vectors is crucial to have an accurate separation.

In the SVM method, only samples which are located on support vectors are important in learning and the algorithm is not sensitive to other points. However, it is important to have enough samples to find the best support vectors. One way to find appropriate support vectors is to estimate the distance between defined boundaries and support vectors (points that lie on the borders of the margins) and finally choosing the boundary that has the maximum distance with both classes. Two possible hyperplanes are shown in Figure 2.4 based on different support vectors. The black hyperplane better separates two classes comparing to green hyperplane, since the margin is maximum.

When the data set is not linearly separable, the algorithm can use a kernel function to map the points into a higher dimensional space where datapoints become linearly separable and finding the support vectors is possible. The kernel function chosen affects the model performance. Some popular kernel functions are Radial Basis Function, Polynomial and Linear kernels.

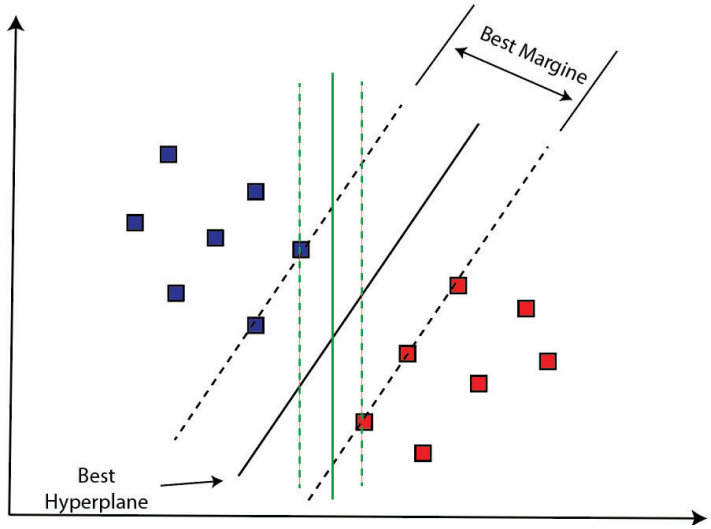

Figure 2.4: SVM Hyperplane.

Basic SVM method was generally designed to separate two classes. Therefore, in order to classify 13 categories of calls, we need to use the multi-class SVM method. In this method, the problem is solved by dividing it into multiple binary classification cases. Then, the final decision is made using the integration of decisions obtained from multiple binary classifications. There are two different methods to divide multi-classification problem into multiple binary classification cases. The first one is called one versus one and it breaks down the problem into finding the decision boundaries for each pair of classes. The second approach is called one versus all. In this approach, the multi-classification procedure is decomposed into the problem of finding the decision boundaries for each class against all other classes. In this study, we use one versus all approach which gave us a better result compared to the other method.

2.5.2 K Nearest Neighbors (KNN)

KNN is another supervised learning algorithm that can be used for classification purposes. In this algorithm, a new sample is classified based on a majority voting mechanism using K neighborhood samples in the training data set (Cover and Hart 1967). In fact, training samples with labels in feature space divide this space into different parts. Once categories of points are defined, the following steps are used to classify a new data point:

1. New sample is mapped to feature space
2. First K closest samples to the new sample are selected
3. New sample will be allocated to the class which has more samples in the K selected samples.

Different methods can be used to measure the distance between samples such as Euclidean, Manhattan, or Makowski distance. Values of K can also be different according to the problem.

Figure 2.5 shows the effect of different K values on classification. If K=1 then new sample will be assigned to class blue. If K=5 then sample will be assigned to class red because in the 5 closest samples, red samples outnumber blue samples.

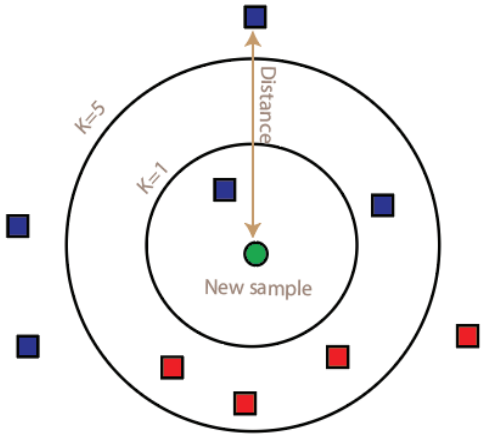

Figure 2.5: KNN classification example.

2.5.3 Fully connected classifier network

The second part of Alexnet consists of fully connected layers that build the classifier part of the network. Fully connected layers consist of nodes, called neurons. These neurons are connected to each other using links. Each connection has a weight that is adjusted during the learning process by minimizing a loss function that is the quantity indicating the error between predicted output and desired output. In this study we tried to train the classifier part of the Alexnet using the features extracted from the feature extraction part. Figure 2.6 shows the architecture of the classifier we used that is the same as the Alexnet fully connected part except for the output layer. Since we have 13 categories in our data set, we set the number of neurons in the output layer to 13 while Alexnet has 1000 neurons in output layer. The network was trained for 150 epochs. Each epoch is a single round of training with all samples in the training data set. The "Adam

optimizer" method was utilized as the learning function that is an optimization technique used to find the minimum loss value. The loss function was categorical cross entropy and the equation of the function is defined as follows:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{M} y_{i,c}\log\left(p_{i,c}\right) \tag{2.1}$$

Where y is either 0 or 1 showing the true label, p is the predicted probability, c is the number of categories and i is the number of samples.

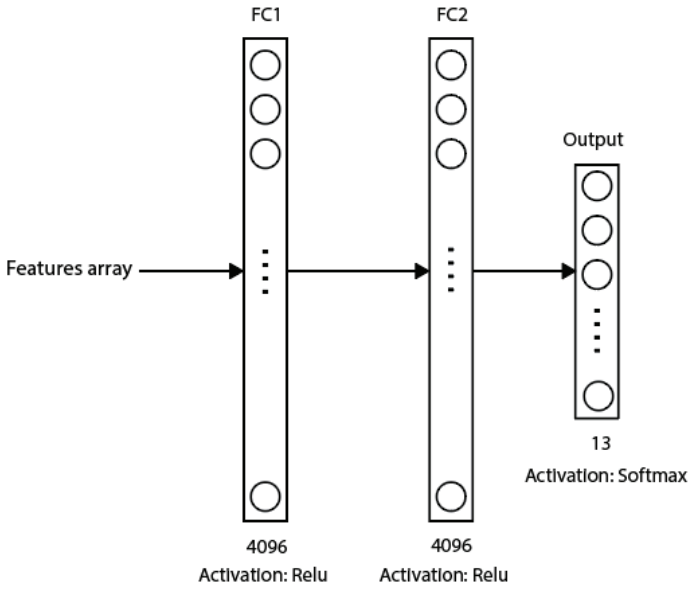

Figure 2.6: Fully connected classifier network.

2.6 **Data partitioning for classifiers validation**

In order to analyze the accuracy of classification methods, a k-fold cross validation method was used. In this method, the data is partitioned into k random folds. One of the folds is selected as the test set and the other folds are assigned to the training set. Then, this process is repeated until each of the divided folds has been selected as the test set once. The final accuracy value is

determined by averaging all accuracies measured for each fold (Arlot and Celisse 2010). The advantage of using k-fold cross-validation is that the classification method can be validated on the whole data set. A value of k=10 was chosen in this study which means the data is divided into 10 folds.

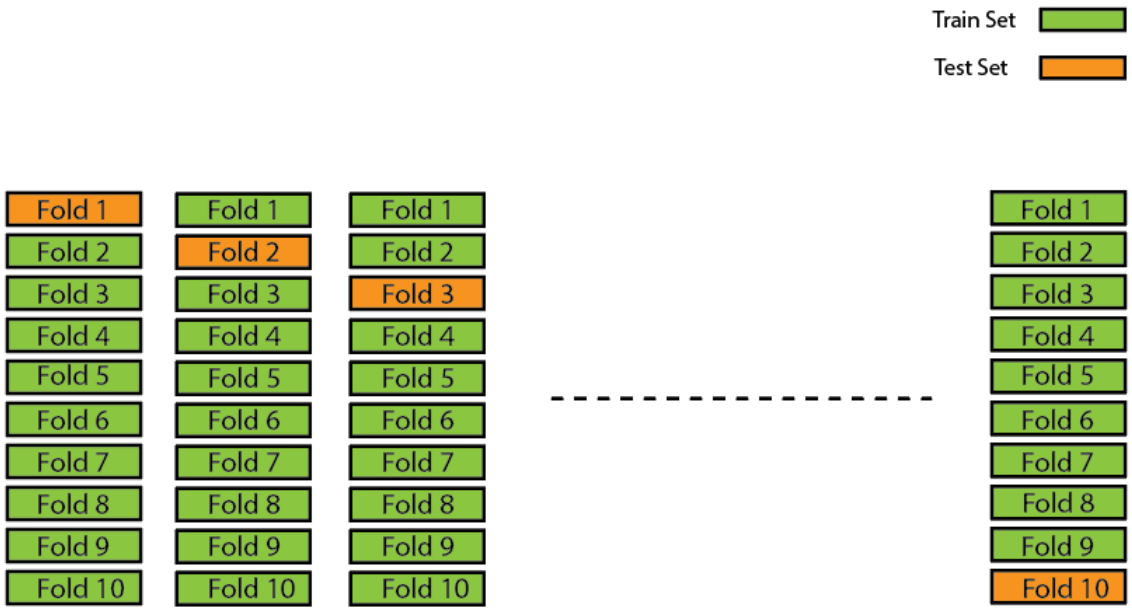

Figure 2.7: K-fold cross validation

Training the model requires additional considerations when there are not enough samples for some categories of calls. This problem can result in poor classification accuracy in classes with low samples. Undersampling is one of the techniques that can be used to overcome class-imbalanced data problems. This strategy removes samples in the larger classes to obtain training data sets with balanced numbers of samples in different classes. In our case, first the data is divided into train and test set using the K-Fold cross validation method and then, only in the training set, classes with numbers of samples over 200 were undersampled to 200. This reduced the effect of imbalanced data sets on the training of the model as shown in figure 2.8. The undersampling was

repeated 10 times for each fold to have a fair approximation of the undersampled categories and the final result was the average of all the results for 10 times undersampling.

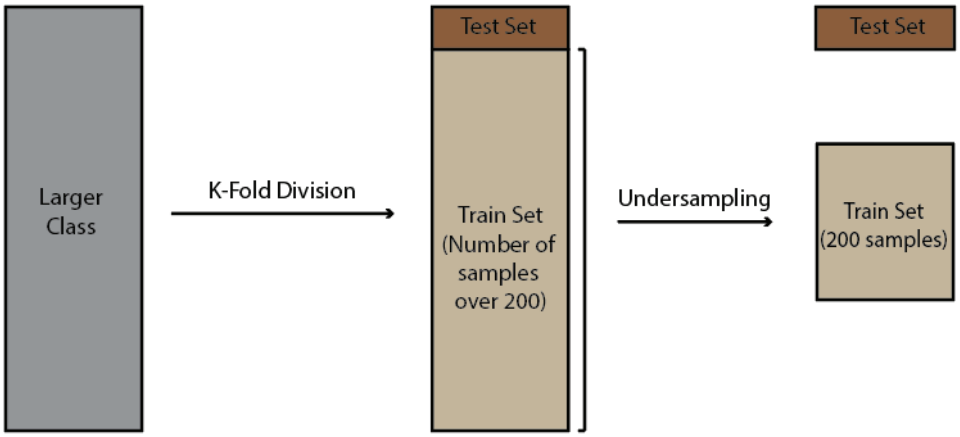

Figure 2.8: Under-sampling after cross validation

2.7 **Generating Dendrogram of Call Categories Using D-prime Score**

We used a D-prime score to estimate the distance between categories. The D-prime score is a measure that is derived from Signal Detection Theory (Peterson, Birdsall et al. 1954) and is used to estimate the distance between signal and noise (see Figure 2.9). If both distributions are Gaussian, D' is the number of standard deviations between their means. It can also be used in classification problems as discriminability score between two categories.

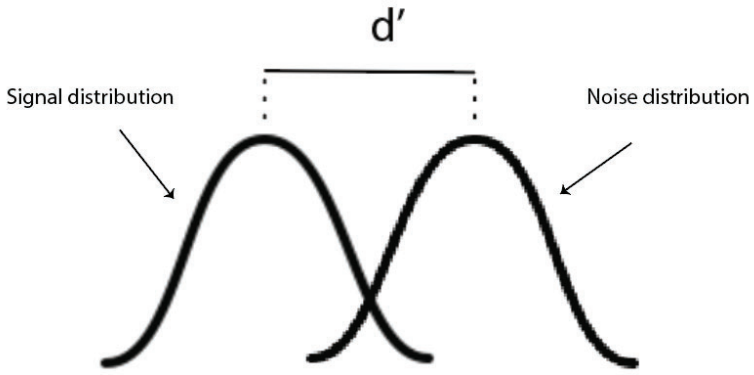

Figure 2.9: D-prime measure for to signals.

In order to calculate the D-prime score, first we divided the classification problem into multiple binary SVM classification problems for all pairs of categories. Then, we estimated a 2 by 2 confusion matrix for each pair after applying SVM. To classify each pair, we did the 10-Fold cross validation and undersampled the larger class to the number of samples in the smaller class. In the next step we produced a dendrogram based on the average D-prime score between categories in each pair. In order to produce the dendrogram, we used Unweighted Pair Group Method with Arithmetic Mean (UPGMA) algorithm, available in the BMEToolbox (http://www.bioinformatics.org/mbetoolbox/) (Cai, Smith et al. 2006).

Chapter 3: Results

3.1 **Denoising autoencoder results**

In order to increase the accuracy of analysis, we trained a denoising autoencoder on the spectrograms using 2000 good quality calls. The autoencoder was trained for 15 epochs. Mean square error was the loss function and Adam optimizer was used as the learning function in the process of training the network. Figure 3.1 shows the training and validation loss of the denoising autoencoder. The output of the denoising autoencoder on naturally contaminated vocalizations is shown in Figure 3.2. In order to assess the impact of the denoising autoencoder on our analysis, we ran our classifier with and without denoising the data set and the confusion matrix for both is shown in section 3.4.

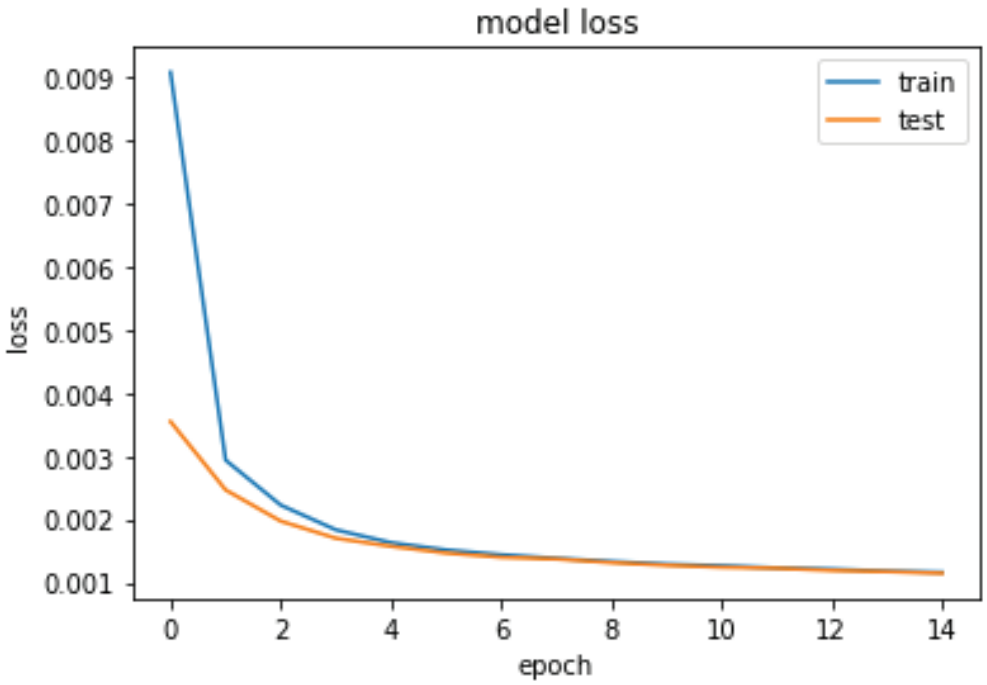

Figure 3.1: Training and test loss of the autoencoder.

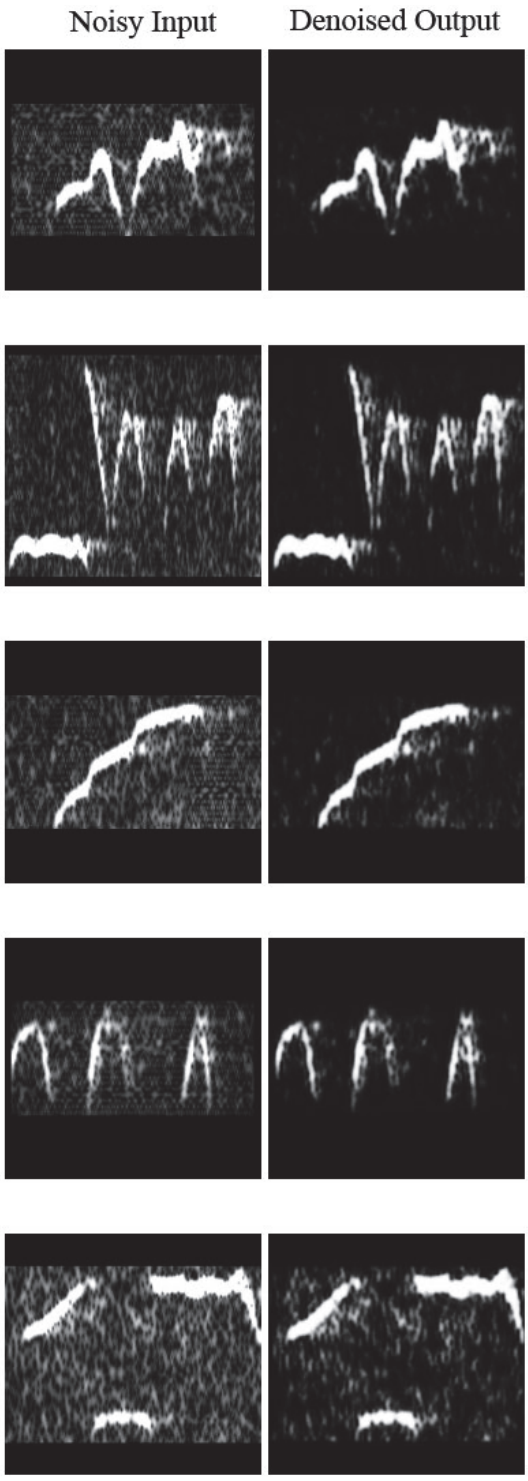

Figure 3.2: Results of denoising autoencoder. The left column shows the inputs and right column shows the denoised output.

3.2 **Classification results**

In this Section we will compare different classification methods that were applied on features extracted using Alexnet including SVM, KNN, and a fully connected neural network. In order to analyze the performance of these classification algorithms, different evaluation metrics have been used including Accuracy, Precision, Recall and F1-Score. Accuracy is the ratio of the number of correct predictions to all of the predictions. Since we have many pairs of categories with imbalanced data sets, accuracy may not be sufficient to analyze the performance of the model. Therefore, we used other metrics as well. To better understand these metrics, consider an example category such as "Complex". Recall shows how many of samples in the "Complex" category were correctly classified as "Complex". Precision shows how many of samples classified as "Complex", truly belong to "Complex" category. F1-Score considers both recall and precision and is a harmonic mean of these two metrics. The formulas of the evaluation metrics can be described as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

$$F1\_Score = \frac{2 * (precision * recall)}{precision + recall} \tag{3.3}$$

$$Accuracy = \frac{TP + TN}{TF + FN + TN + FP} \tag{3.4}$$

where,

- **TP**: True Positive. Samples that belong to positive class and are correctly classified as positive class.
- **FP**: False Positive. Samples that belong to positive class and are incorrectly classified as negative class.
- **TN**: True Negative. Samples that belong to negative class and are correctly classified as negative class.

  FN: False Negative. Samples that belong to negative class and are incorrectly classified as positive class.

Table 3.1 shows the performance of different classification algorithms as measured by F1-Score, Precision, Recall and Accuracy. Each method has specific parameters that need to be tuned in order to achieve the best performance. For KNN, the best performance was obtained for k=10. We also trained a fully connected neural network with 3 layers (first layer: 1024 neurons, second layer: 256 neurons, output layer: 13 neurons). SVM showed the highest accuracy comparing to all other methods. The overall accuracy was 63.67% at C=10 and γ=10-4. The best parameters were obtained by an exhaustive search of all combination of C and γ, using values of 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10 and 100 for both C and γ. The kernel that was used in SVM was a radial basis function.

Table 3.1: Comparison of different classifiers on features extracted using transfer learning.

Classifier	Parameters	Accuracy	F1-Score	Precision	Recall
SVM	C=0.1 $\gamma=10^{-4}$	49.81%	50.06%	64.07%	49.81%
SVM	C=1 $\gamma=10^{-4}$	63.06%	64.20%	70.04%	63.06%
SVM	**C=10** **$\gamma=10^{-4}$**	**63.67%**	**65.17%**	**70.43%**	**63.67%**
SVM	C=100 $\gamma=10^{-4}$	63.02%	64.68%	70.12%	63.02%
KNN	K=8	49.09%	49.79%	64.37%	49.09%
KNN	K=10	49.34%	49.86%	64.63%	49.34%
Alexnet Fully Connected Classifier	-	61.55%	63.78%	68.62%	61.55%

3.3 **Detailed analysis of SVM classification**

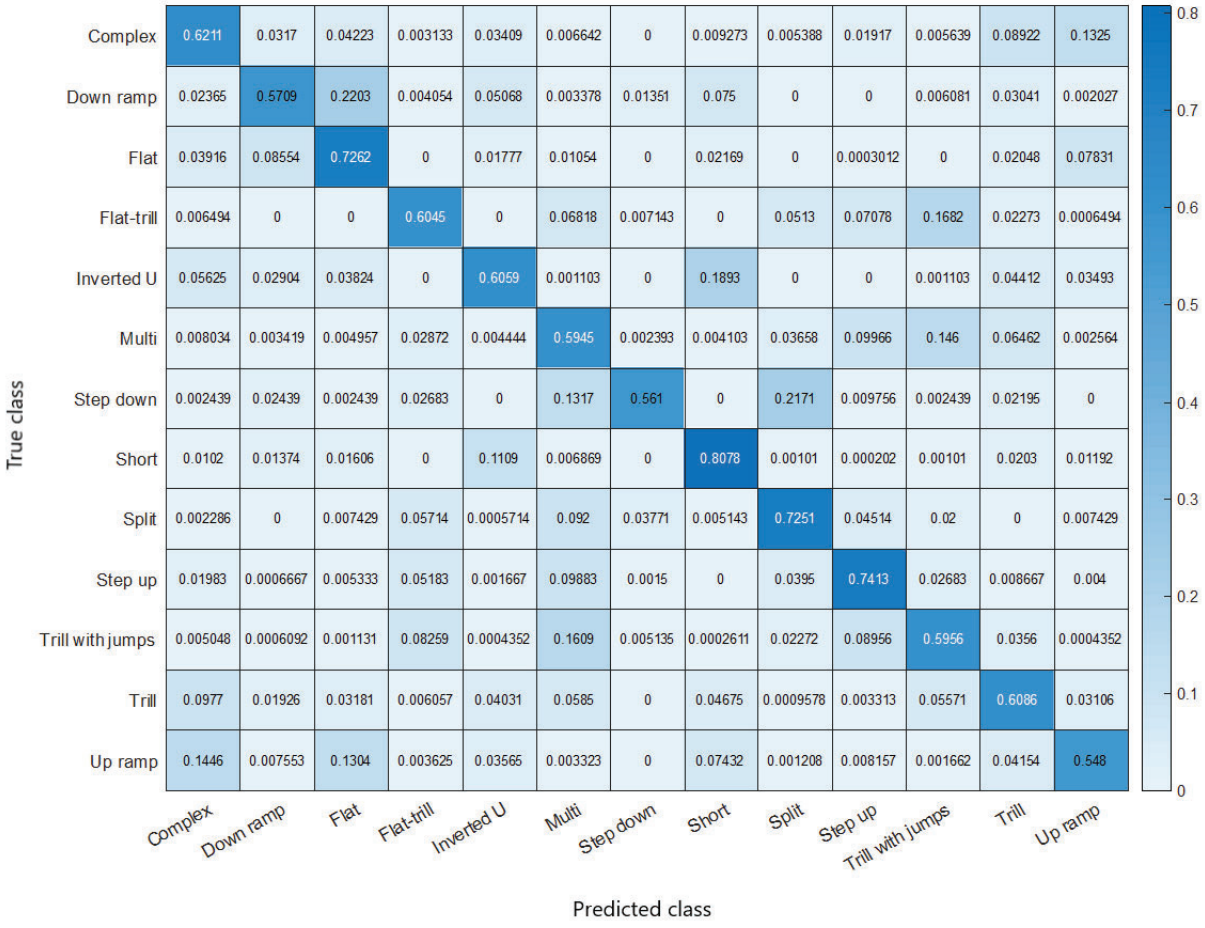

Figure 3.3: Confusion matrix of SVM classifier.

In order to analyze the performance of SVM for each category, a confusion matrix is presented in Figure 3.3. The confusion matrix provides a more detailed evaluation of classification since it is possible to see misclassification rates between pairs of categories. The labels on the left side of the matrix show true categories and the labels on the bottom of the matrix show predicted categories using the classifier. For each row (i.e., true category), the cell on the matrix diagonal shows the number of correct predictions and other cells show the predictions that were incorrectly assigned to other categories. In order to normalize these numbers, cells on each row were then

divided by all the predictions in that row. Looking at the data, one can see, for example, that 62.11% of the complex calls were correctly classified as complex while 13.25% were misclassified as "Up ramp," the category most likely to catch the miscategorized complex calls. The short call category has the highest correct classification rate, around 80.78% and the lowest correct classification rate belongs to "Up ramp," which was correctly identified at a rate of 54.8%. For all other categories the rate is above 55%. Table 3.2 shows the performance of the SVM method based on the F1-Score, Precision and Recall. As reported in this table, the classifier shows acceptable performance for all categories. The average F1-Score, Precision and Recall of the classifier is above 63%.

Table 3.2: Classification performance of the SVM based on F1-Score, Precision and Recall**.**

Category	F1-Score	Precision	Recall
Complex	54.01%	48.01%	62.18%
Down ramp	41.64%	34.24%	56.6%
Flat	54.51%	44.06%	72.4%
Flat-trill combination	43.1%	33.87%	60.65%
Inverted U	42.43%	33.28%	60.09%
Multi	47.73%	40.17%	59.39%
Step down	60.67%	65.92%	59.21%
Short	75.99%	71.98%	80.75%
Split	62.87%	57.03%	73.38%
Step up	70.54%	67.55%	74.07%
Trill with jumps	62.48%	65.89%	59.62%
Trill	72.93%	91.09%	60.84%
Up ramp	55.34%	56.43%	54.56%
Average	65.17%	70.44%	63.66%

In order to visualize samples in feature space based on features extracted from AlexNet, we used the t-Distributed Stochastic Neighbor Embedding (t-SNE) method. t-SNE, is a dimensionality reduction method that is useful for visualizing high-dimensional data. Figure 3.4

depicts all vocalization samples and can help us to see how data clusters are scattered in feature space. The figure shows that most categories have some distance from most other categories. However, each category has an overlap with one other category.

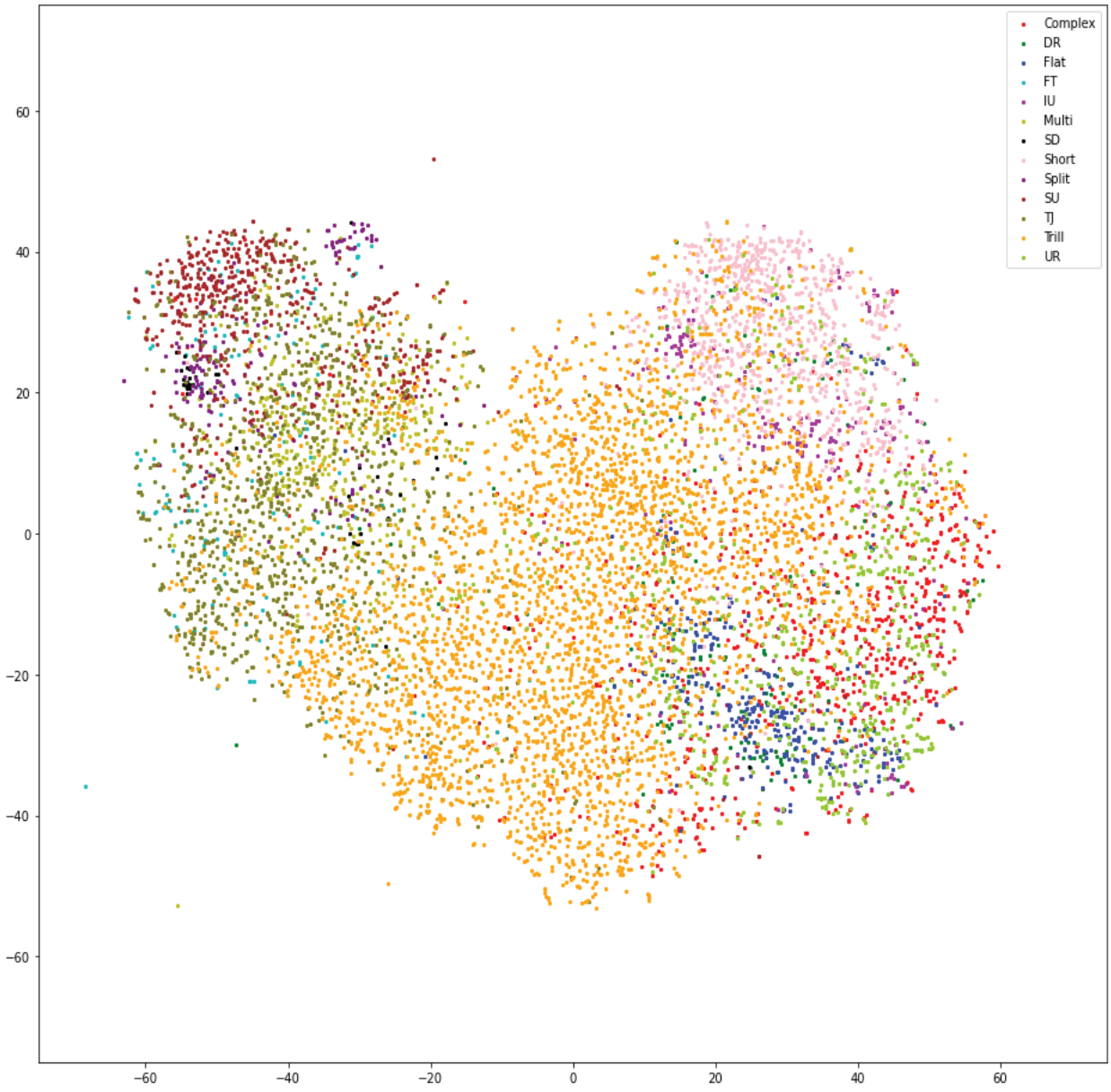

Figure 3.4: t-SNE visualization of features extracted from Alexnet network.

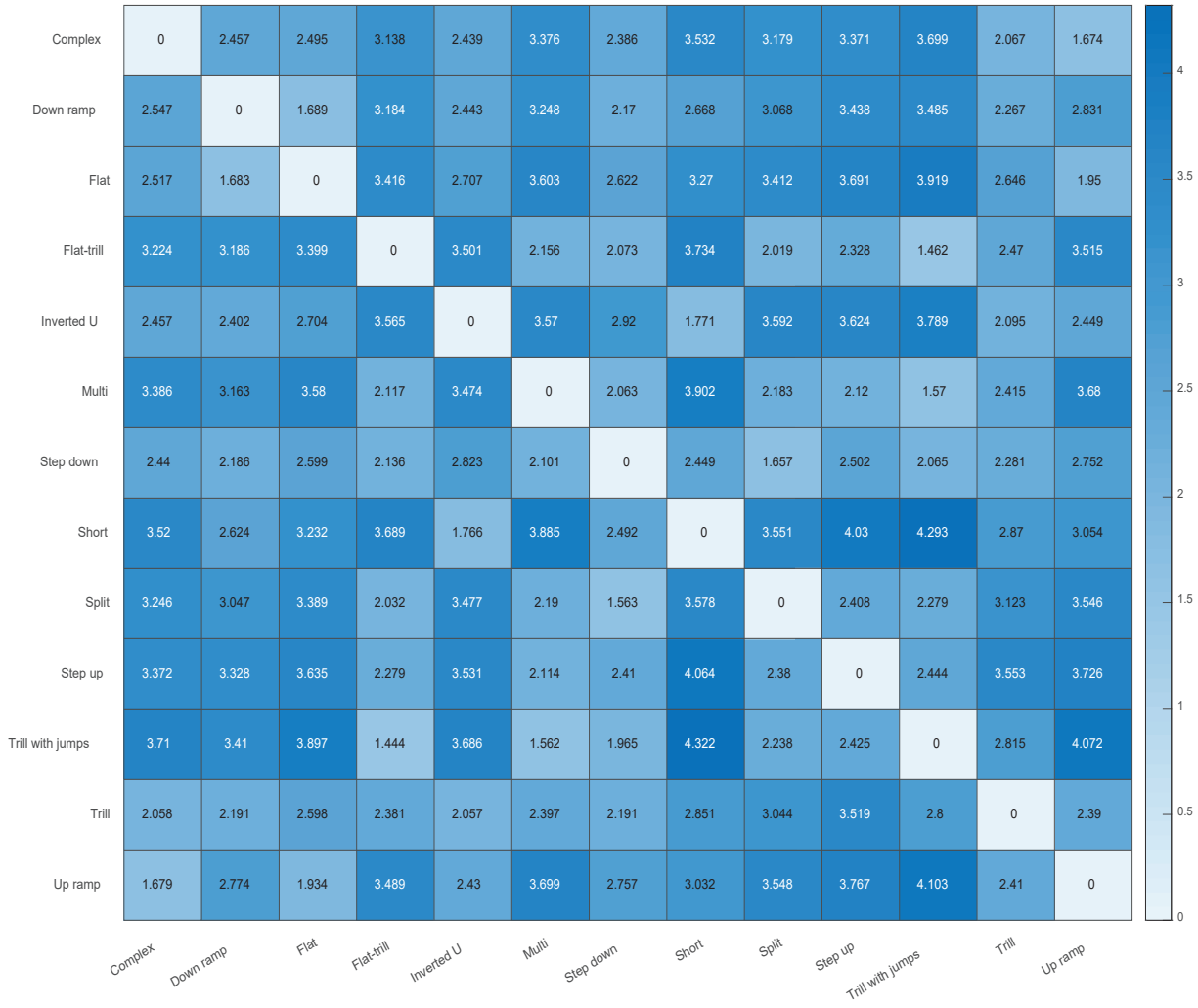

Figure 3.5: D-prime score for the pairwise classification.

As mentioned in chapter 3, we used D-prime score to measure the distance between pairs of categories using 2 by 2 confusion matrix obtained from each pairwise classification. The parameters of the SVM model were the same as the parameters that were used for the multi-classification problem. Figure 3.5 depicts a matrix that contains the D-prime score for the categories. The names on the left side of matrix shows the categories that were considered as positive class and the name on the bottom of the matrix shows the categories that were considered as negative class. Figure 3.6 shows the dendrogram generated from the average D-prime scores

between pairs of categories. We cut the dendrogram at the level which D-prime score equals 2 (see Figure 3.6) assuming that categories above this level are separate enough and extracted 7 new categories as below:

- Category 1: “Up ramp” and “Complex’
- Category 2: “Trill”
- Category 3: “Flat” and “Down ramp”
- Category 4: “Short” and “Inverted U”
- Category 5: “Trill with jumps”, “Multi” and “Flat-trill combination”
- Category 6: “Step up”
- Category 7: “Split” and “Step down”

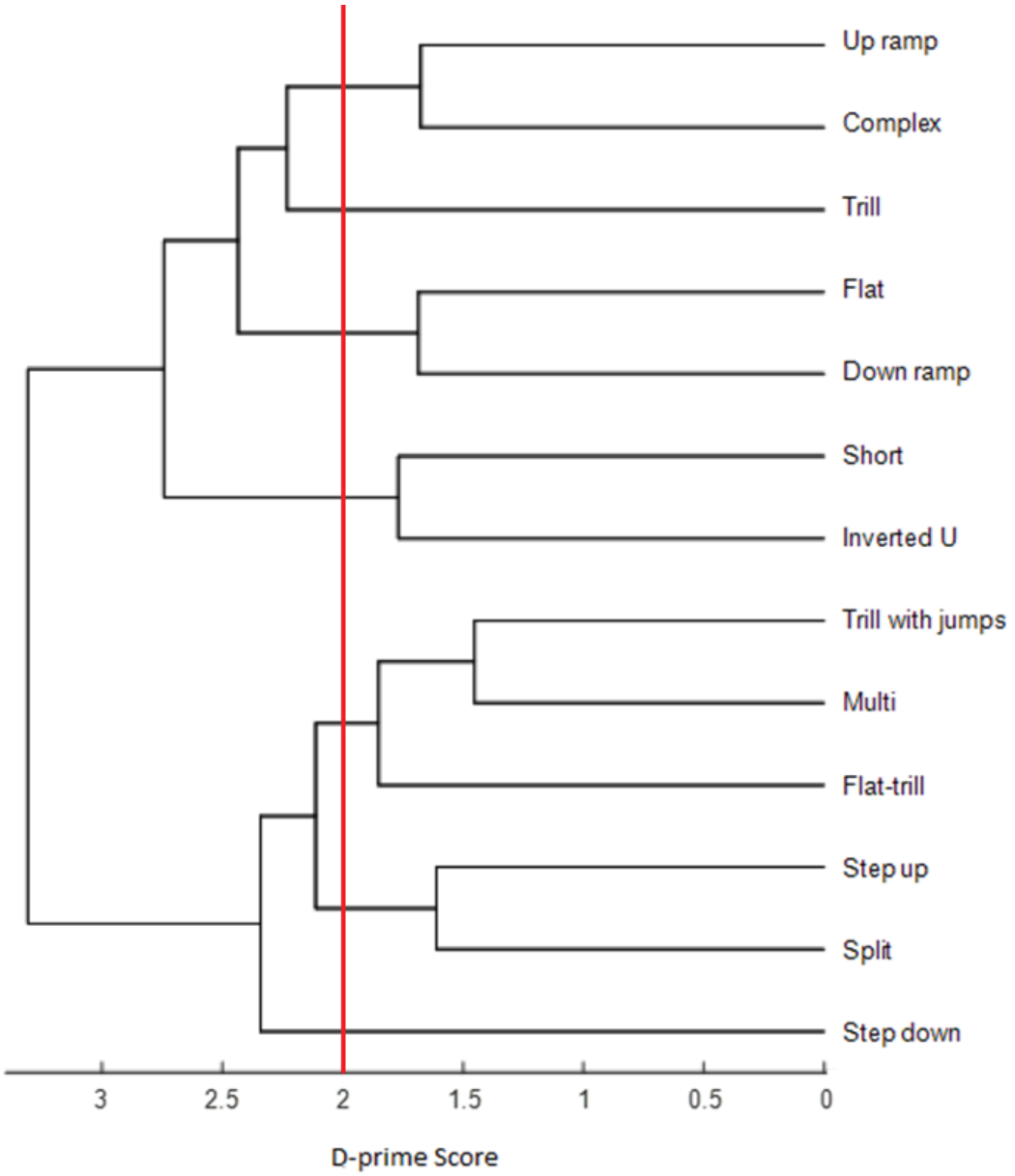

Figure 3.6: Dendrogram of call categories. The red line shows the cut-off level that was used to extract new categories.

In the next step, we tried to train another SVM classifier based on the 7 categories obtained from the dendrogram to analyze the separability of them. In order to train the classification model, categories with number of samples over 500 were undersampled to 500 and we achieved the accuracy of 77.8% using 10-fold cross validation. Figure 3.7 shows the confusion matrix of the

SVM classifier. According to the confusion matrix, the correct classification rate for all the groups is above 0.7 except for "group 1" which is 0.69. Also, note that with only two exceptions, no call is miscategorized as any other call more than 10% of the time. The one exception is category 5, which is misclassified as category 6 in 12.5% of the cases. The main reason is that category 5 includes "Split" calls which are similar to "Multi" calls in Category 6. The other exception is category 7, which is misclassified as category 7 in 12.6% of the cases. Table 3.3 shows more detailed performance evaluation based on F1-Score, Precision, Recall and Accuracy.

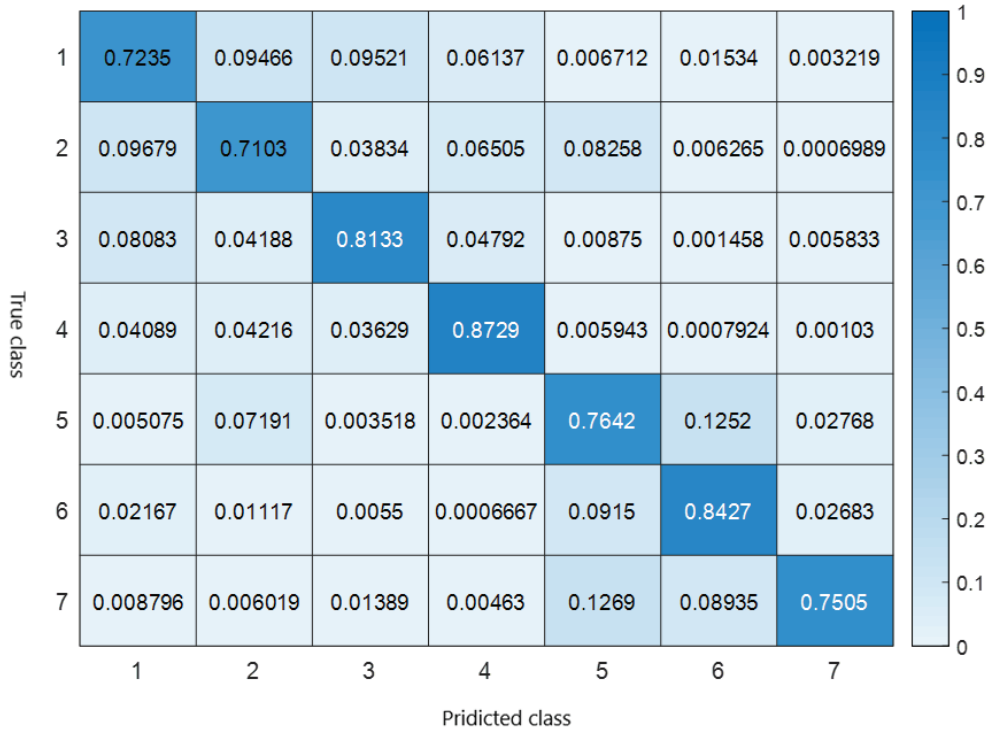

Figure 3.7: Confusion matrix of SVM classifier on new 7 categories.

Table 3.3: Classification performance of the SVM on the 7 new categories based on F1-Score, Precision and Recall**.**

Categories	F1-Score	Precision	Recall
1	70.27%	68.36%	72.38%
2	78.92%	88.84%	71.02%
3	64.11%	53.01%	81.60%
4	80.60%	74.91%	87.30%
5	76.05%	75.81%	76.50%
6	72.71%	64.34%	84.22%
7	71.11%	68.16%	75.26%
Average	76.07%	77.90%	75.77%

In order to visualize the overlap between categories we used linear discriminant analysis (LDA). This approach helps us to project the features into 1-dimension and plot the distribution of samples according to linear discriminant (LD) vector. It should be noted that we reduced the number of features to 128 using Principal Component Analysis (PCA) before applying LDA. The reason for this is that the projection did not work properly when we had large number of features. Figure 3.8 shows 12 representative examples of pairs of categories projected onto the principle LD axis.

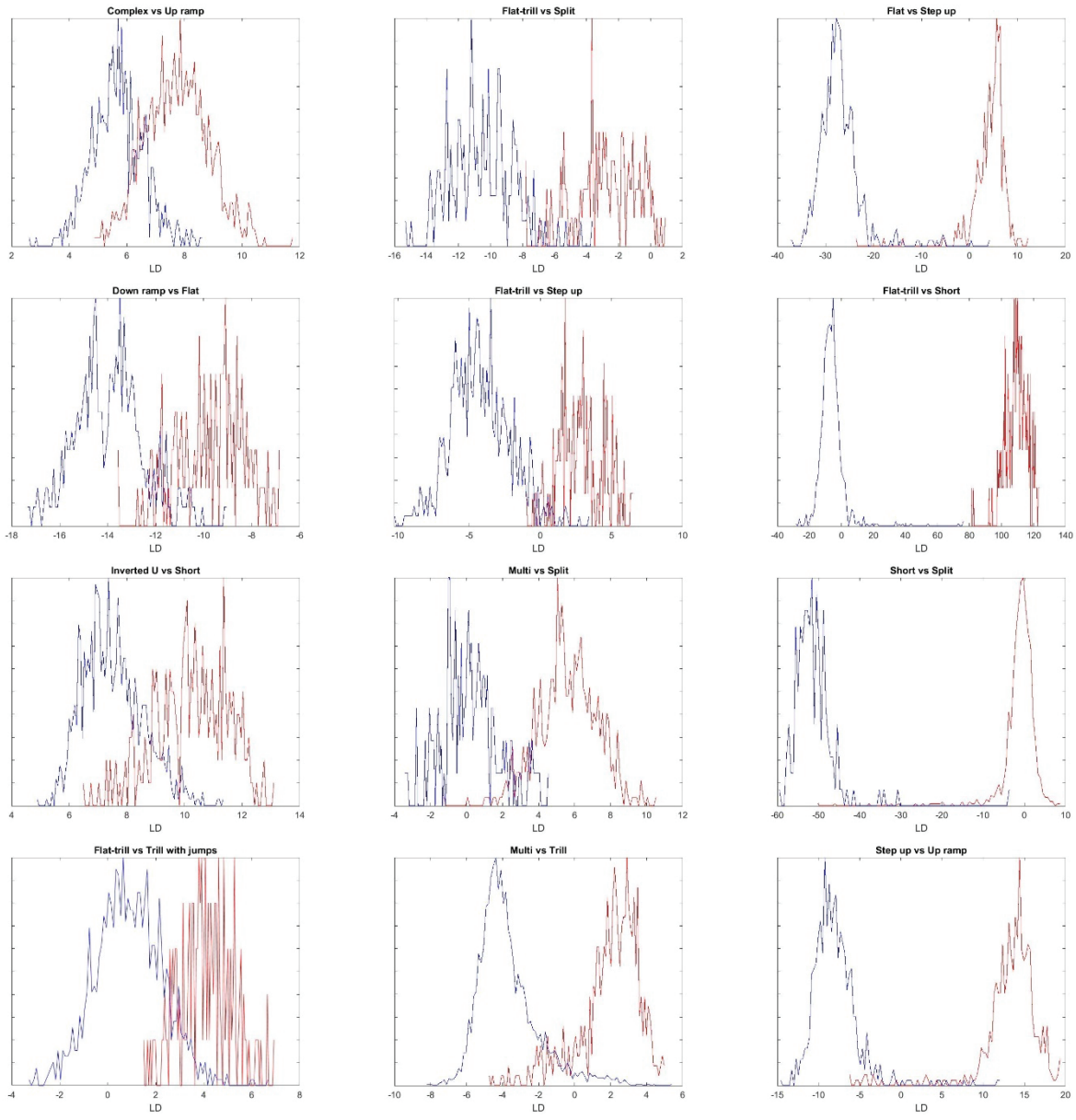

Figure 3.8: Sample distributions for pairs of call categories after projection to one dimension using linear discriminant analysis. The left column shows the top 4 overlapped category pairs. The middle column shows 4 pairs with a medium level of overlap and the right column shows the 4 pairs with the largest separation. SVM confusion matrix was used to find out the overlap between categories.

3.4 **Effects of De-Noising on Classification Performance**

The confusion matrix in Figure 3.9 is presented to evaluate the effect of the denoising autoencoder on classification performance. This confusion matrix shows the performance of SVM classifier trained on spectrograms that are not denoised using the autoencoder. By looking at the confusion matrix, we can see that the accuracy is lower than the SVM classifier that was trained on spectrograms that were denoised using the autoencoder. The highest effect was on "Down ramp" and "Step down" categories.

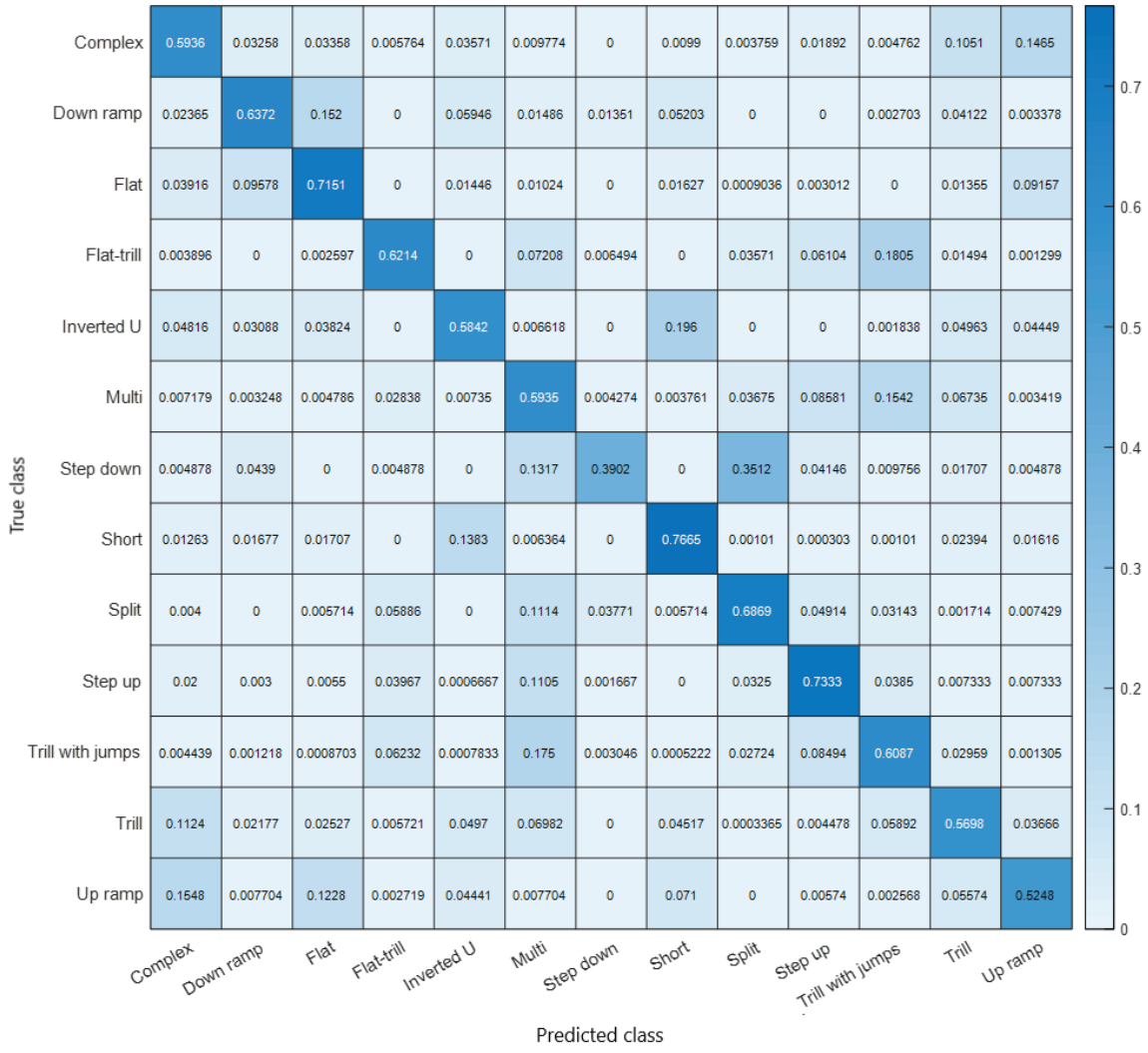

Figure 3.9: Confusion matrix of the SVM classifier without Denoising Autoencoder.

3.5 **Evaluation of Deepsqueak Classifier**

As mentioned in chapter 1, Deepsqueak is a software package for detection, classification and clustering of rats' vocalizations. In order to evaluate the performance of Deepsqueak, we applied its classifier on our data set. The classifier is a CNN which is trained on 11 categories instead of the 13 in our dataset. Therefore, we excluded two other categories, "Trill with jumps" and "Multi," from our data set. Another difference in this classifier is that one of the categories is called "Complex Trill" which is not included in Wright et al. (2010), categorization scheme. However, there is a "Flat Trill" category in Wright et al. categorization scheme which is the most similar category to "Complex trill". Therefore, we substituted the label "Flat Trill" for "Complex trill" in our performance evaluation data set.

The accuracy of Deepsqueak on our dataset was 52.96%. Figure 3.10 shows the confusion matrix of the Deepsqueak classifier on our data set. The confusion matrix shows that Deepsqueak performs poorly in detecting categories such as "Down ramp", "Inverted U", "Split" and "Trill". However, the correct classification rate is high for categories such as "Complex Trill", "Short" and "Step Down". As shown in Table 3.4, the overall performance of Deepsqueak was poor on our data set based on the average F1-Score, Precision and Recall.

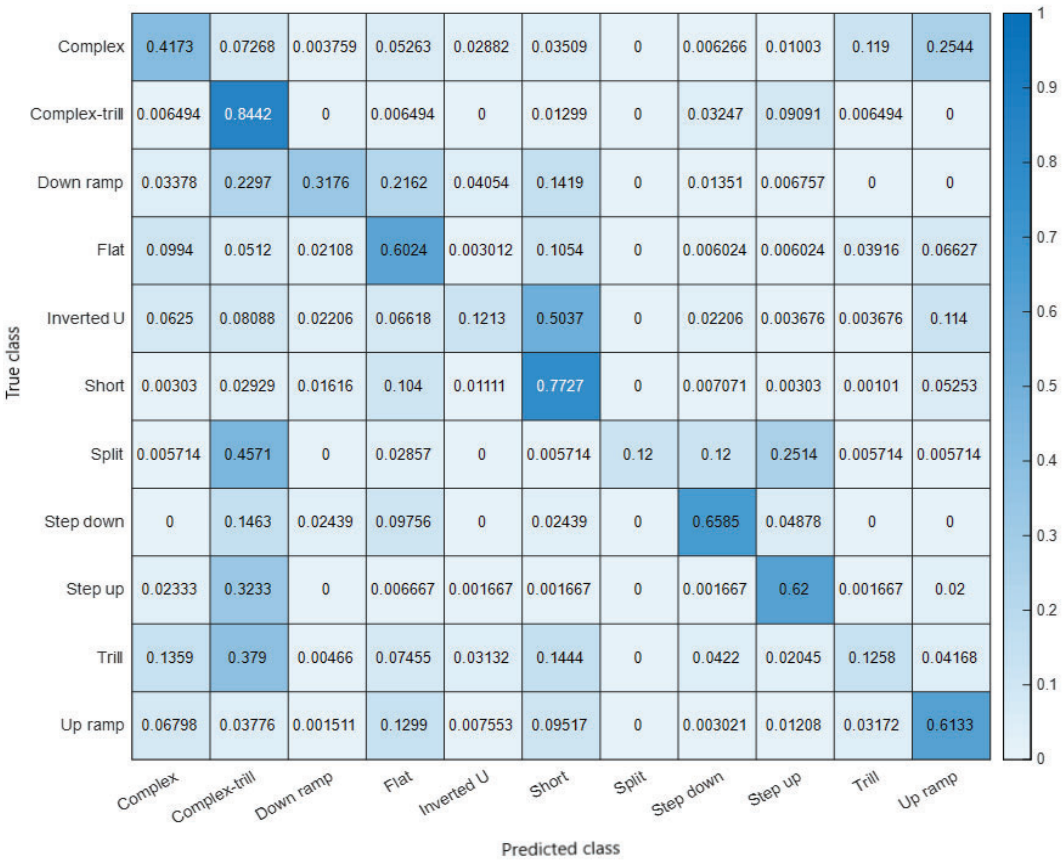

Figure 3.10: Confusion matrix of Deepsqueak.

Table 3.4: Classification performance of the Deepsqueak based on F1-Score, Precision and Recall.

Category	F1-Score	Precision	Recall
Complex	55.11%	50.06%	61.84%
Down ramp	20.12%	18.95%	22.62%
Flat	35.82%	25.62%	61.06%
Flat-trill combination	31.09%	24.24%	45.47%
Inverted U	32.01%	22.45%	56.14%
Multi	35.05%	25.92%	55.04%
Step down	39.28%	52.53%	37.73%
Short	67.45%	66.24%	69.13%
Split	53.87%	48.70%	62.56%
Step up	66.55%	66.58%	66.92%
Trill with jumps	55.40%	54.87%	56.29%
Trill	55.25%	82.81%	41.54%
Up ramp	43.36%	46.85%	40.67%
Average	53.02%	62.37%	51.21%

3.6 **Efficacy of Manually Selected Features for Classification**

In this section manually selected features proposed in Deepsqueak were analyzed. These features include shape, frequency, and duration that were briefly explained in the introduction. In order to investigate the effectiveness of using manually selected features, we trained a SVM classifier using these features and performed a performance evaluation on the classifier. According to the test results presented in the Figure 3.11 and Table 3.5, the classifier cannot properly detect the calls based on the manually selected features. The correct classification rate for six categories is under 40% and it shows that manually selected feature extractors cannot extract appropriate feature from the calls.

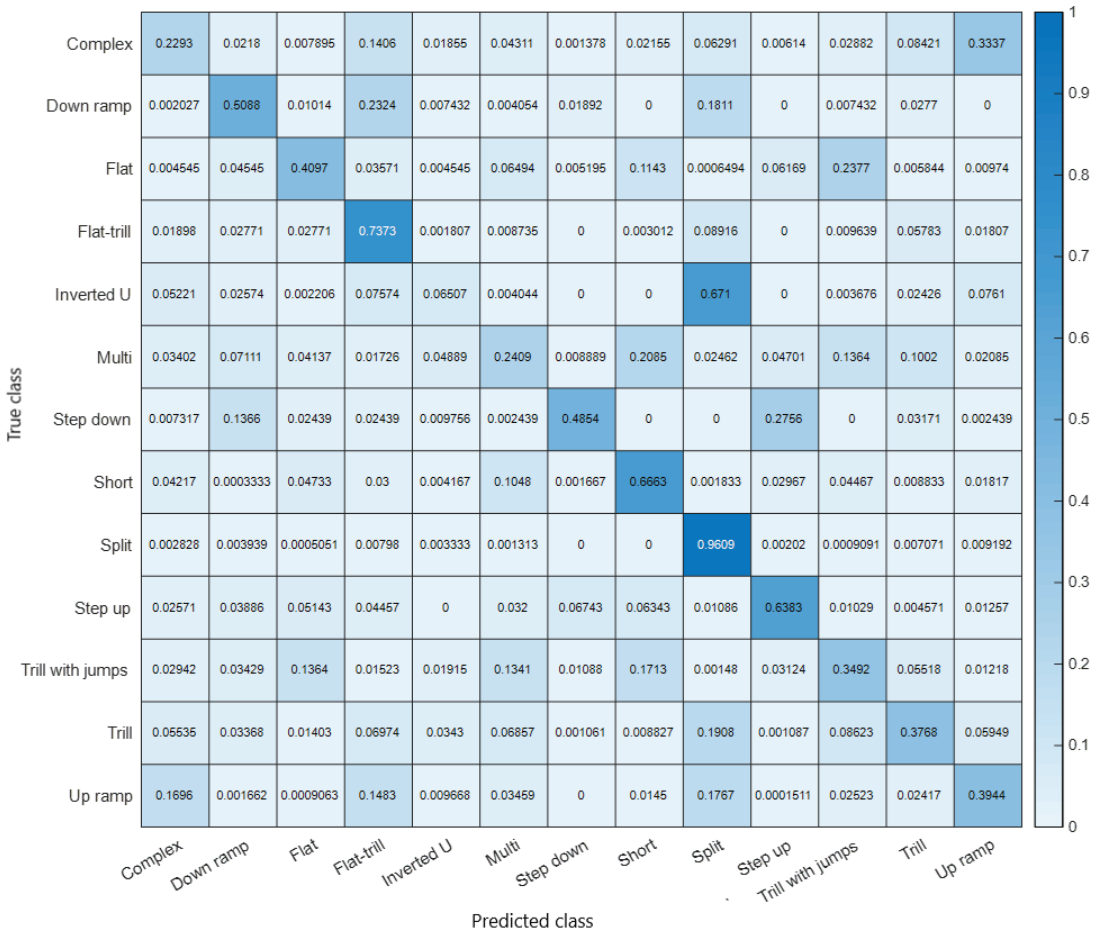

Figure 3.11: Confusion matrix of the SVM classifier using manually selected features.

Table 3.5: Classification performance of the SVM on manually selected features based on F1-Score, Precision and Recall.

	F1-Score	Precision	Recall
Complex	25.64%	29.67	23.02%
DR	30.50%	22.03%	51.44%
Flat	24.22%	17.66%	40.68%
FT	41.03%	28.63%	73.84%
IU	6.96%	7.79%	6.59%
Multi	21.84%	20.19%	24.27%
SD	38.13%	36.97%	46.42%
Short	56.450%	49.30%	66.72%
Split	61.22%	44.97%	96.11%
SU	55.94%	50.71%	64.35%
TJ	38.59%	43.56%	34.92%
Trill	52.20%	85.28%	37.67%
UR	34.69%	31.33%	39.40%
Average	44.36%	55.29%	44.28%

In the next analysis, we tried to add the middle frequency of each call to the features extracted from the Alexnet and trained an SVM classifier based on this combined feature set. In order to estimate the middle frequency, we used the middle of coordinate of a box drawn around each call. Figure 3.12 shows the confusion matrix of the classifier. The confusion matrix is almost identical to the SVM classifier based on the Alexnet features alone (Figure 3.3) showing that adding middle frequency to the features does not change the performance of the classifier and correct classification rate remains relatively constant.

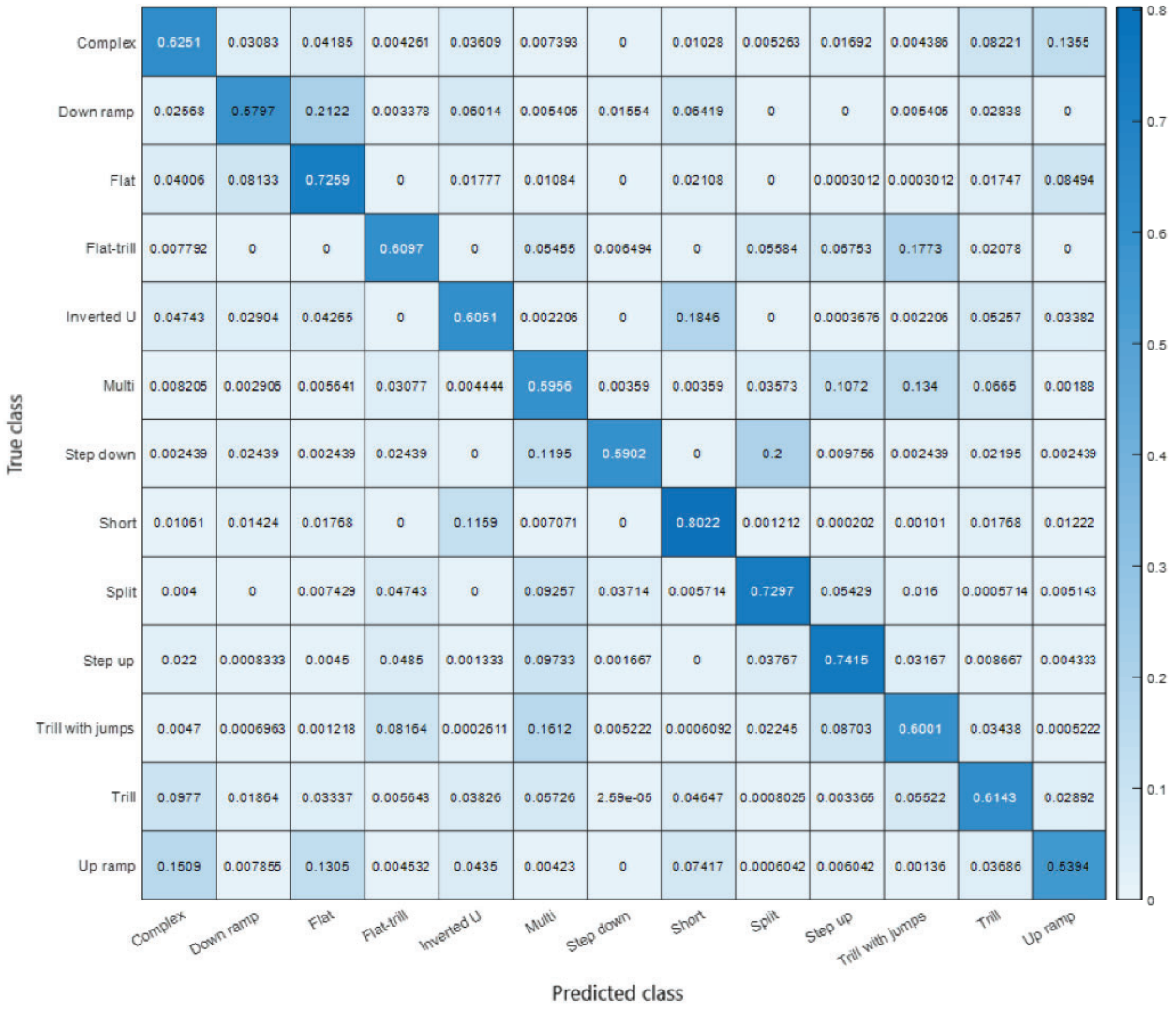

Figure 3.12: Confusion matrix of the SVM classifier using combined Alexnet features and frequency.

3.7 **Automated Clustering**

In this study we also used the k-means clustering method to automatically identify separate clusters of calls from features extracted using Alexnet. The most important thing in clustering a data set is to define the number of clusters. In order to find the optimal number of clusters, we attempted to use the elbow method. In this method, first a graph is plotted whose y-axis is “the sum of squared distances of data points to their nearest cluster center” and the x-axis is “changes

in the number of clusters" as shown in Figure 3.13. The point where curve bends, shows the optimal number of clusters. According to Figure 3.13, a bend in the curve occurs around k=8. Therefore, we set the number of clusters to 8 for k-means clustering. In order to investigate the output clusters, we decided to see what portion of these clusters belong to different human defined categories as shown in Table 3.6. According to proportions presented in the Table 3.6, there are several categories which have a high degree of overlap with manual categories. The table can be summarized as follows:

- cluster 1: Mostly trill.
- Cluster 2: Overlaps with Complex, Up ramp and Trill.
- cluster 3: Mostly Trill and Trill with jumps.
- Cluster 4: Overlaps strongly with short calls, but some contamination with trills.
- Clusters 5: Mostly short but some contamination with trill with jumps.
- Cluster 6: Combination of Trill with jumps, Trill, Step up and Multi.
- Cluster 7: Mostly Trill.
- Cluster 8: Mostly Trill with jumps.

Figure 3.14 shows the t-SNE visualization of samples after clustering them using k-means and it indicates that the clustering algorithm is not able to find identify well-isolated clusters. Note that categories with the samples more than 200 were undersampled before clustering. The primary reason for this analysis is to determine if automatic clustering discovers the same categories as humans. If we have a huge number of trills, the automatic clustering will not work properly. Every cluster will be some part of the trill category. So, the only way to get a fair assessment of whether automatic clustering can work is to reduce the imbalanced data effect.

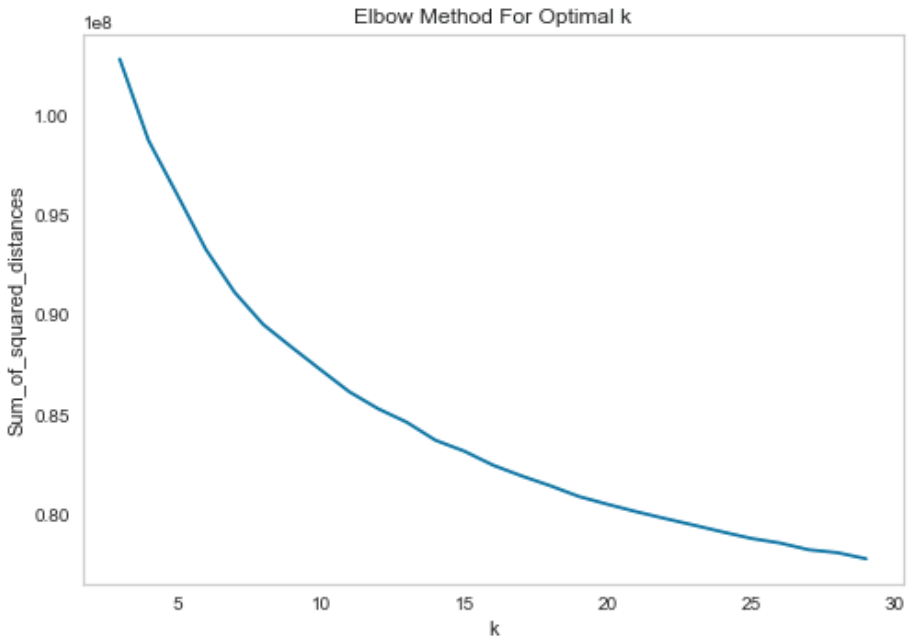

Figure 3.13: Elbow method for optimal K.

Table 3.6: Relationship between k-means clustering and human defined labels.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Complex	8.91%	36.90%	0.71%	3.41%	2.25%	0.57%	0.48%	5.86%
Down ramp	2.01%	4.18%	0.00%	1.81%	0.00%	0.76%	0.00%	1.01%
Flat	6.79%	11.88%	0.00	2.31	0.00	0.28	0.00	0.43
Flat-trill combination	0.00%	0.08%	3.37%	0.00%	6.75%	1.98%	7.19%	0.14%
Inverted U	1.26%	3.68%	0.00%	9.59%	0.00%	0.28%	0.00%	0.65%
Multi	1.01%	0.25%	4.61%	0.55%	5.69%	32.61%	12.10%	2.60%
Step down	0.10%	0.00%	0.35%	0.00%	1.32%	2.36%	0.12%	0.07%
Short	1.76%	0.59%	0.00%	47.04%	0.13%	0.57%	0.12%	0.22%
Split	0.10%	0.08%	0.35%	0.05%	9.79%	5.86%	3.95%	0.00%
Step up	0.55%	1.26%	0.53%	0.10%	51.19%	14.93%	2.51%	0.22%
Trill with jumps	0.55%	0.50%	29.61%	0.10%	21.16%	23.16%	64.43%	1.45%
Trill	69.15%	12.97%	60.46%	28.11%	1.32%	15.88%	9.10%	85.31%
Up ramp	7.80%	27.62%	0.00%	6.93%	0.40%	0.76%	0.00%	2.03%

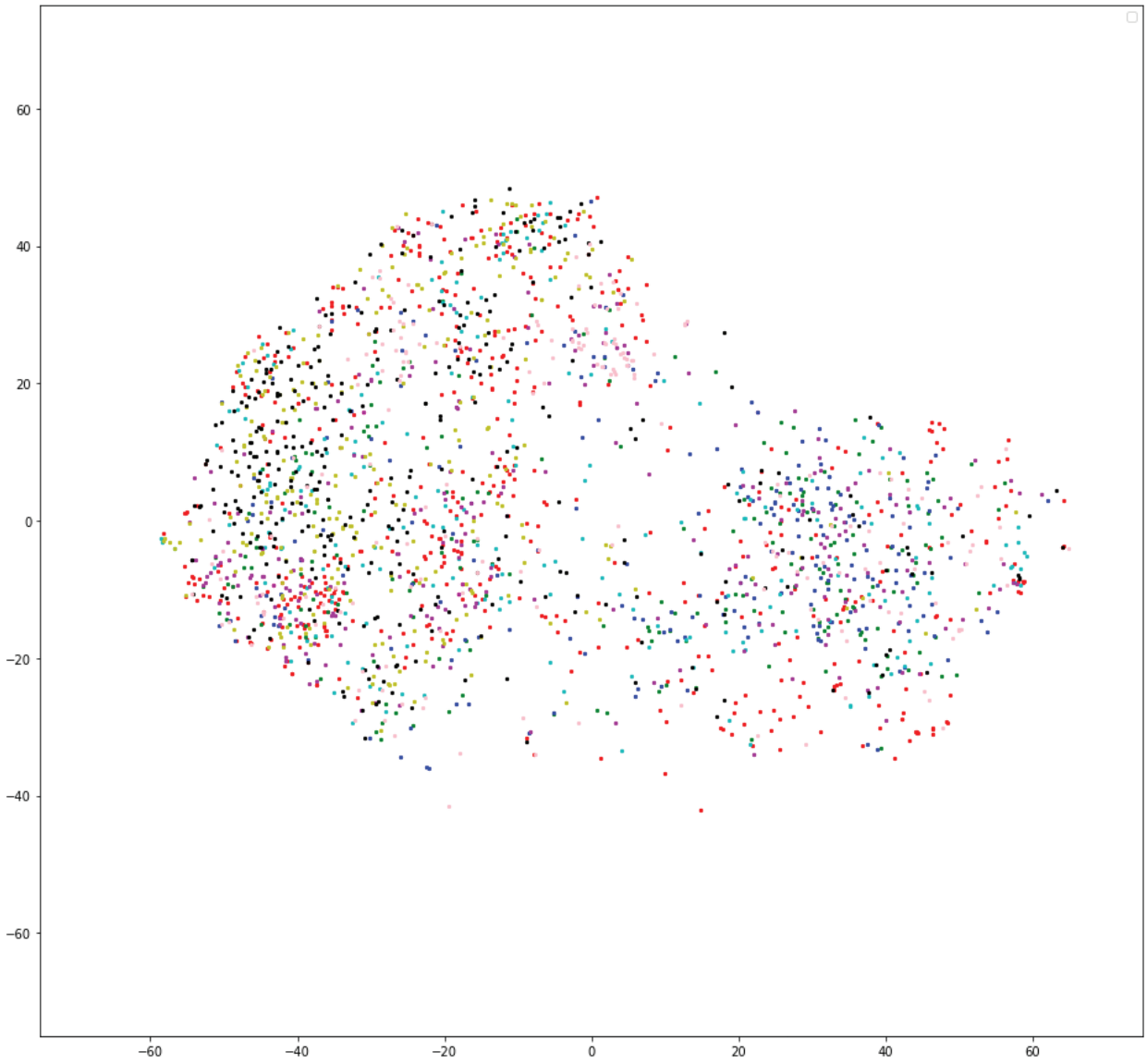

Figure 3.14: t-SNE visualization of features extracted from Alexnet network based on the labels extracted by k-means algorithm.

Chapter 4: Discussion

4.1 General discussion

Our primary goal in this thesis was to determine which previously identified calls are similar and which are distinct based on computational features. Therefore, we used Alexnet to extract features from spectrograms of the calls. Since we did not have enough samples to train a complete neural network, we decided to use a transfer learning method that means we used a pretrained neural network for feature extraction. After extracting features, we utilized different classification algorithms to investigate the separability of calls. The best performance was achieved by the SVM method, and our results showed that the algorithm can classify calls with an average accuracy of 63.67%. The confusion matrix from this analysis showed that several call categories were highly similar. As these are difficult for a machine to discriminate, they may also be difficult for rats to tell apart. Based on the similarities of some of these calls, we suggest that calls might be better grouped into 7 separate categories, plus a "Composite" category that is composed of several different calls combined. Whether these 8 categories are used differently by rats remains to be tested.

Another question addressed in this thesis was whether the 14 categories identified by Wright et al. (2010) are, in fact, the correct way to classify the calls. In other words, do the categories identify in their analysis correspond to natural clusters with clear separation between them? To address this, we applied a fully automated clustering algorithm, k-means, to our collection of calls. We used an elbow method to define the optimal number of clusters and then tried to analyze these clusters by comparing them to human made categories. Table 3.6 shows the portions of each cluster that belong to human made categories and according to the table we could not find a strong correlation between machine generated clusters and our original human-generated categories. Therefore, automated clustering doesn't find the same categories as humans and

doesn't fit the data well. The poor fit is evident in the t-SNE visualization of the 8 clusters identified by k-means depicted in Figure 3.14. In this plot, there are no distinct clusters which might suggest a natural category.

Previous attempts to categorize calls have been based on manual inspection of the calls' spectrograms. The weakness of this approach is that it is highly subjective, and the categorization results vary between different people, especially for a subset of very similar calls. Our scores are derived from a deep neural network trained to categorize visual objects. As such, it is less likely to be subject to human-induced bias. We also trained a classifier on manually selected features to investigate usefulness of these features and the sorting accuracy was poorer than that achieved using features extracted from the CNN according to the Figure 3.11 and Table 3.5. There are two caveats regarding the use of manually selected features. One of the caveats is that the proper segmentation of the call's profile (i.e., the plot of the frequency corresponding to the highest intensity peak at each frequency) is not possible according to the low signal to noise ratio of the recorded calls. Another caveat is that our method of extracting the profile, proposed in Coffey's DeepSqueak software, is poor at tracing profiles in trills due to their rapid fluctuations and lower intensity.

The data from this thesis helps address the question of how many categories of calls should be used in the analysis of rat vocalizations. Based on the basic categorization scheme, calls are grouped into two categories of flat and frequency modulated (Burgdorf, Kroes et al. 2008, Wöhr, Houx et al. 2008, Ciucci, Ahrens et al. 2009). Brudzynski et al, divided the frequency modulated category into 4 separate categories (Brudzynski 2009). Recently, a more detailed categorization scheme has proposed that breaks down the calls into 14 categories (Wright, Gourdon et al. 2010). In this study, we analyzed Wright et al. categorization scheme using features extracted

automatically by a machine learning algorithm. Note that we excluded "Composite" category in all our analysis, since it is a combination of different calls. Therefore, we made a 13 by 13 matrix containing the D-prime score obtained from pairwise SVM classification for the rest of 13 categories and generated a dendrogram using this matrix. Based on dendrogram, we concluded that branches of the tree above a cut-off level of 2 are distinct enough to be considered as "real" categories. Finally, based on this cut-off level, we concluded that these 13 categories should be grouped into 7 categories. Therefore, our conclusion is that calls might be better separated into 7 categories, plus "Composite," resulting in 8 separate categories in total. We applied the SVM classifier on the 7 extracted categories and the overall accuracy was 77.44 showing that these 7 categories are highly discernible.

Ultimately, the categories used for rat vocalizations should be based on both functional and physical characteristics. Unfortunately, very little is known about the functional characteristics of rat vocalizations. Takahashi et al (Takahashi, Kashino et al. 2010) divided 50 kHz calls into two clusters of 40 kHz and 60 kHz and suggested that 40 kHz calls are related to feeding and 60 kHz calls are related to walking. Wöhr, Houx et al. (2008) suggested that 50 kHz flat calls are produced when the rat is separated from its mate. It has been also proposed that rats emit flat calls during nape attacks (Burke, Kisko et al. 2018). Burke et al. found that trill and trill with jumps are associated with slow locomotion (walking) and composite, split and multi are related to fast locomotion (running and jumping) (Burke, Kisko et al. 2017). They proposed a dendrogram that clusters call categories based on their functional characteristics as shown in figure 4.1A. By comparing this dendrogram with the dendrogram that we generated using computational features of the calls (Figure 4.1B) we do not find significant agreement between these two figures. For example, in Figure 4.1A we can see that Trill with jumps and Trill and are grouped together, but

in the Figure 4.1B Trill with jumps is next to the multi. These results show that more research is needed before we have a clear picture of the functional characteristics of each of the possible call categories. At this point, we may have to rely on physical similarity of the calls to define the categories.

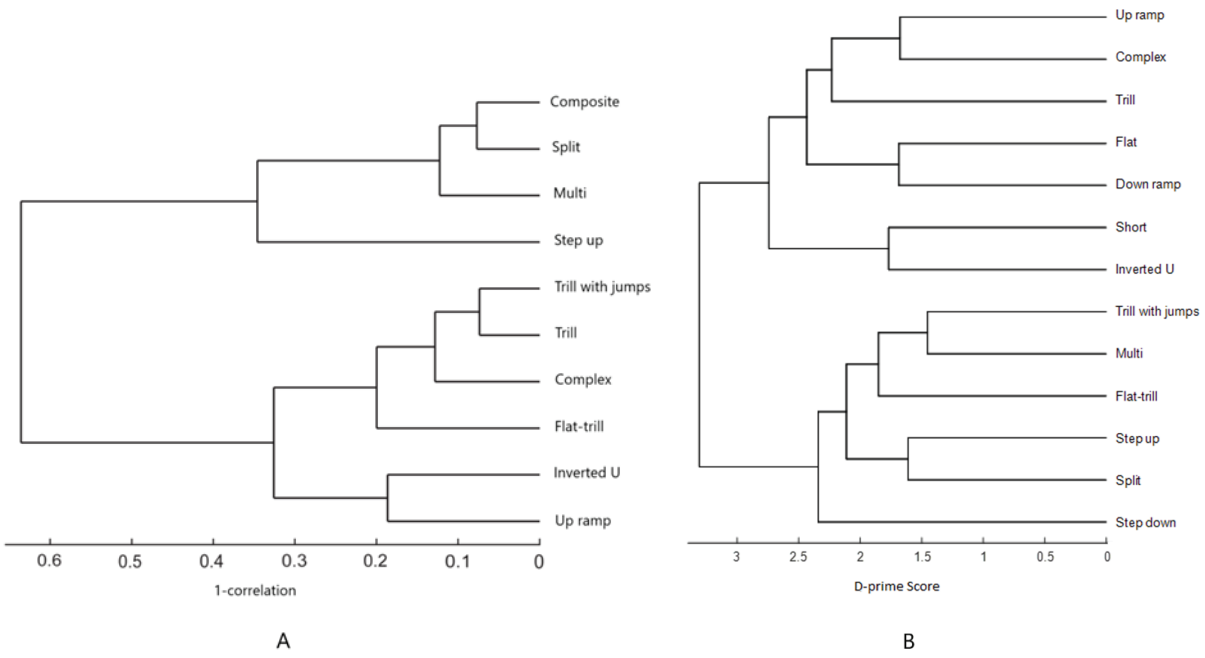

Figure 4.1: Dendrograms of call categories. A) Based on their behavioral correlates from Burke et al. (2017). B) Based on computational features used in this thesis (see Figure 3.6).

An underlying question is whether rat calls are categorical. According to the t-SNE visualization of the features in the Figure 3.4, each category has an overlap with at least one other category, even though it is distant from several other categories. Therefore, we don't see separate clusters, and this raises the issue of whether calls may be a continuum. However, it is difficult to say with certainty that there are no separate categories. In fact, not being able to define hard boundaries between acoustically distinct calls doesn't not certainly mean that the rat doesn't perceive boundaries. For example, categorical perception is seen for several phonemes such as /ba/ and /pa/. While the voice-onset time used to discriminate these phonemes varies continuously,

human perception snaps rather suddenly from one category to the other somewhere in the middle of the range. (Wood 1976). Furthermore, the quality of recorded data could affect the analysis due to low signal-to-noise ratio because of the quietness of calls, the microphone distance, and noise from shuffling of feet or rustling of bedding. Therefore, future research may be needed to better understand the correct categorization of the calls.

Although the purpose of this project was not to derive a method for others to use to sort calls, but rather to study the separability of call categories that have been proposed by others, it is informative to compare our sorting accuracy with that reported by others. We also tested the Deepsqueak (Coffey, Marx et al. 2019) classification model on 11 categories of our dataset and the accuracy was 52.96% which is not high enough for studies of the functional characteristics of different calls. Note that direct comparison of Deepsqueak with our model is not fair, since our model is trained on part of the same data set upon which it is tested. Nevertheless, our method could easily be extended to categorize new calls. Acoustilytix (Ashley, Snyder et al. 2021) also has the ability to classify calls but it has been only trained on 3 categories for 50 kHz calls that are a composite of Wright et al.'s 14 proposed categories and it is mentioned in the paper that the overall accuracy is in the range 71% to 78%. The reliability human observer labeling is also an issue for categorization. In an earlier study from our lab, two human raters categorize the same data independently and the agreement between them was 36% (Burke, Kisko et al. 2017). The problem was that they confused "Trills", "Complex", "Composite" and "Trill with jumps". So, to human observers, these calls are highly similar. On the rest of the calls, the agreement was 98%.

4.2 **Future Direction**

One of the obstacles in reaching higher accuracy in this this study was the low number of samples in some categories. However, the accuracy can be improved by having a large number of samples for each category. Neural networks show higher accuracy comparing to other algorithms when the number of training samples increases. Therefore, by having a large number of samples in the training set, it is possible to train a neural network to achieve better performance. Another problem in reaching higher accuracy is the low signal to noise ratio of the recorded calls. In some cases, the amplitude of the signal is not clearly discernible from the background, and this could cause poor classification accuracy. One way to increase the quality of calls could be using multiple microphones and then applying noise cancelation techniques. The other approach could be finding new ways to locate the microphone at a closer distance to the rats leading to background noise reduction.

References

Ågmo, A. and E. M. Snoeren (2015). "Silent or vocalizing rats copulate in a similar manner." PLoS One **10**(12): e0144164.

Allen, T. A., et al. (2007). "Single-unit responses to 22 kHz ultrasonic vocalizations in rat perirhinal cortex." Behavioural brain research **182**(2): 327-336.

Allin, J. T. and E. M. Banks (1972). "Functional aspects of ultrasound production by infant albino rats (Rattus norvegicus)." Animal Behaviour **20**(1): 175-185.

Anderson, J. W. (1953). "The production of ultrasonic sounds by laboratory rats and other mammals." Science.

Arlot, S. and A. Celisse (2010). "A survey of cross-validation procedures for model selection." Statistics surveys **4**: 40-79.

Ashley, C. B., et al. (2021). "Acoustilytix™: A Web-Based Automated Ultrasonic Vocalization Scoring Platform." Brain Sciences **11**(7): 864.

Barfield, R. J. and L. A. Geyer (1972). "Sexual behavior: ultrasonic postejaculatory song of the male rat." Science **176**(4041): 1349-1350.

Bishop, C. M. (2006). Pattern recognition and machine learning, springer.

Blanchard, D. C., et al. (2005). "Defensive responses to predator threat in the rat and mouse." Current protocols in neuroscience **30**(1): 8.19. 11-18.19. 20.

Brown, R. E. (1979). "The 22-kHz pre-ejaculatory vocalizations of the male rat." Physiology & behavior **22**(3): 483-489.

Brudzynski, S. M. (2005). "Principles of rat communication: quantitative parameters of ultrasonic calls in rats." Behavior genetics **35**(1): 85-92.

Brudzynski, S. M. (2009). "Communication of adult rats by ultrasonic vocalization: biological, sociobiological, and neuroscience approaches." Ilar Journal **50**(1): 43-50.

Burgdorf, J., et al. (2000). "Anticipation of rewarding electrical brain stimulation evokes ultrasonic vocalization in rats." Behavioral neuroscience **114**(2): 320.

Burgdorf, J., et al. (2001). "Nucleus accumbens amphetamine microinjections unconditionally elicit 50-kHz ultrasonic vocalizations in rats." Behavioral neuroscience **115**(4): 940.

Burgdorf, J., et al. (2008). "Ultrasonic vocalizations of rats (Rattus norvegicus) during mating, play, and aggression: Behavioral concomitants, relationship to reward, and self-administration of playback." Journal of comparative psychology **122**(4): 357.

Burke, C. J., et al. (2018). "Do juvenile rats use specific ultrasonic calls to coordinate their social play?" Animal Behaviour **140**: 81-92.

Burke, C. J., et al. (2017). "Avoiding escalation from play to aggression in adult male rats: The role of ultrasonic calls." Behavioural processes **144**: 72-81.

Burke, C. J., et al. (2017). "Specific 50-kHz vocalizations are tightly linked to particular types of behavior in juvenile rats anticipating play." PLoS One **12**(5): e0175841.

Cai, J. J., et al. (2006). "MBEToolbox 2.0: An enhanced version of a MATLAB toolbox for Molecular Biology and Evolution." Evolutionary Bioinformatics **2**: 117693430600200002.

Ciucci, M. R., et al. (2009). "Reduction of dopamine synaptic activity: degradation of 50-kHz ultrasonic vocalization in rats." Behavioral neuroscience **123**(2): 328.

Coffey, K. R., et al. (2019). "DeepSqueak: a deep learning-based system for detection and analysis of ultrasonic vocalizations." Neuropsychopharmacology **44**(5): 859-868.

Cortes, C. and V. Vapnik (1995). "Support-vector networks." Machine learning **20**(3): 273-297.

Cover, T. and P. Hart (1967). "Nearest neighbor pattern classification." IEEE transactions on information theory **13**(1): 21-27.

Fonseca, A. H., et al. (2021). "Analysis of ultrasonic vocalizations from mice using computer vision and machine learning." Elife **10**: e59161.

Goodfellow, I., et al. (2016). Deep learning, MIT press Cambridge.

Gould, J. and C. Morgan (1941). "Hearing in the rat at high frequencies." Science.

Green, S. and P. Marler (1979). The analysis of animal communication. Social behavior and communication, Springer**:** 73-158.

Kikusui, T., et al. (2003). "Conditioned fear-related ultrasonic vocalizations are emitted as an emotional response." Journal of veterinary medical science **65**(12): 1299-1305.

Kisko, T. M., et al. (2015). "Are 50-khz calls used as play signals in the playful interactions of rats? III. The effects of devocalization on play with unfamiliar partners as juveniles and as adults." Behavioural processes **113**: 113-121.

Krizhevsky, A., et al. (2017). "Imagenet classification with deep convolutional neural networks." Communications of the ACM **60**(6): 84-90.

Liang, N.-Y., et al. (2006). "Classification of mental tasks from EEG signals using extreme learning machine." International journal of neural systems **16**(01): 29-38.

Otte, D. (1974). "Effects and functions in the evolution of signaling systems." Annual Review of Ecology and Systematics **5**(1): 385-417.

Panksepp, J., et al. (2004). "Regional brain cholecystokinin changes as a function of friendly and aggressive social interactions in rats." Brain research **1025**(1-2): 75-84.

Peterson, W., et al. (1954). "The theory of signal detectability." Transactions of the IRE professional group on information theory **4**(4): 171-212.

Portfors, C. V. (2007). "Types and functions of ultrasonic vocalizations in laboratory rats and mice." Journal of the American Association for Laboratory Animal Science **46**(1): 28-34.

Premoli, M., et al. (2021). "Automatic classification of mice vocalizations using Machine Learning techniques and Convolutional Neural Networks." PLoS One **16**(1): e0244636.

Shin, H.-C., et al. (2016). "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE transactions on medical imaging **35**(5): 1285-1298.

Smith, J. C. (1976). "Responses of adult mice to models of infant calls." Journal of Comparative and Physiological Psychology **90**(12): 1105.

Takahashi, N., et al. (2010). "Structure of rat ultrasonic vocalizations and its relevance to behavior." PLoS One **5**(11): e14115.

Thomas, D. A., et al. (1982). "Male-produced ultrasonic vocalizations and mating patterns in female rats." Journal of Comparative and Physiological Psychology **96**(5): 807.

Thomas, D. A., et al. (1983). "Analysis of ultrasonic vocalizations emitted by intruders during aggressive encounters among rats (Rattus norvegicus)." Journal of comparative psychology **97**(3): 201.

Vogel, A. P., et al. (2019). "Quantifying ultrasonic mouse vocalizations using acoustic analysis in a supervised statistical machine learning framework." Scientific reports **9**(1): 1-10.

Wang, X.-W., et al. (2014). "Emotional state classification from EEG data using machine learning approach." Neurocomputing **129**: 94-106.

Wöhr, M., et al. (2008). "Effects of experience and context on 50-kHz vocalizations in rats." Physiology & behavior **93**(4-5): 766-776.

Wöhr, M. and R. K. Schwarting (2007). "Ultrasonic communication in rats: can playback of 50-kHz calls induce approach behavior?" PLoS One **2**(12): e1365.

Wöhr, M. and R. K. Schwarting (2013). "Affective communication in rodents: ultrasonic vocalizations as a tool for research on emotion and motivation." Cell and tissue research **354**(1): 81-97.

Wood, C. C. (1976). "Discriminability, response bias, and phoneme categories in discrimination of voice onset time." The Journal of the Acoustical Society of America **60**(6): 1381-1389.

Wright, J. M., et al. (2010). "Identification of multiple call categories within the rich repertoire of adult rat 50-kHz ultrasonic vocalizations: effects of amphetamine and social context." Psychopharmacology **211**(1): 1-13.