

Ben-Gurion University
Faculty of Engineering
Department of Bio-Medical Engineering

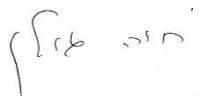
**Identification and classification of ultrasonic
vocalizations in mice using Deep Neural Networks in
order to predict autistic behavior**

Submitted by:

Omri Malbin - 204728513

Yaniv Rotaru - 305143471

Supervisors:



Prof. Hava Golan



Dr. Dror Lederman

Table of contents

Abstract.....	3
Introduction.....	3
USV – Ultrasonic Vocalization.....	3
Goals.....	4
Machine learning and deep learning.....	9
Chapter 1 – Unsupervised learning.....	11
1.1 Methods.....	11
1.2 Preliminary investigation using the MNIST dataset.....	12
1.3 Results.....	13
1.4 Classification of the USV dataset.....	15
1.5 Discussion.....	16
Chapter 2 –Supervised learning.....	17
2.1 Methods.....	17
2.2 Results.....	19
2.3 Discussion.....	24
Chapter 3: active learning.....	25
3.1 Methods.....	25
3.2 Results.....	26
3.3 Discussion.....	27
Summary.....	28
References.....	28

Abstract

Autism is a developmental disorder which is affected by environmental and genetic factors. Mice are considered good model for researching autism because they present rich natural behaviors relevant to the domains of behaviors impaired in autism. One of the phenotyping modalities is the Ultrasonic Vocalizations (USV) which are used to analyze mice communication. Analysis of USV requires to segment the recorded audio signals, i.e., identify the voice regions, and classify the syllables into different categories. This analysis can be used to characterize statistical differences between “healthy” mice and mice with autistic-like behavior, as a basis for developing “diagnostic” tool. Therefore, in this project, we investigated different approaches, namely supervised and unsupervised, for the segmentation and classification of USV signals, based on the deep learning methodologies. First, we developed an unsupervised approach based on autoencoders, in order to reveal the underlying structures of the syllables. This approach yielded an average classification rate of 64%. For the supervised approach, we achieved relatively good performance, i.e., an average classification rate of 96.6%.

Introduction

USV – Ultrasonic Vocalization

Vocal communication exists in several animal species. These species use this communication in different social states like playing, fighting and stressful situations [2]. Vocal communication is particularly relevant for rodent models of neurodevelopmental disorders characterized by social and communication deficits, such as autism and schizophrenia [3]. In mice, the vocal communication is conducted in an ultrasonic range whose frequencies are above the human hearing frequency range (greater than 20 kHz) [3]. Those Ultrasonic Vocalizations (USV) are expressed mainly in the social relations between mother and pups. Therefore, analyzing USV in mice can give us a measure of the social behavior between them [2]. Analysis of USV recordings can provide a great deal of information about the social and mental state of the animals, as well as motor function [4]. Recently, analysis of USV has been used with the aim of expanding existing knowledge about the diseases related to neurology. For example, USV analysis was performed in a model of mice suspected of having Down Syndrome

- these mice had additional chromosome 16 which is analogous to trisomy of chromosome 21 in humans, a condition that cause Down syndrome [5].

Another potential application of USV analysis is in assessing Autism Spectrum Disorders (ASD) or autism-like behavior in animal models. ASD is a lifelong, developmental disability that affects how living creatures communicates with the environment, and how they experience the world around them. ASD in humans is currently diagnosed based on a series of behavioral assessments. The challenge for researchers is to try and uncover the biological basis for these typical behaviors in order to improve diagnosis and identify potential targets for treatment [1]. One of the autism diagnostic tools is evaluation of impaired communication.

Using pup's USV, for example, can demonstrate the importance of USV analysis - a relatively lower rate of USV may be an indication for impaired communication, while a relatively higher rate of USV may indicate that the mice are cold or isolated from the nest. Therefore, when using ASD model in pups, a basic test that can be performed is to separate the puppy from the mother and monitor its readings [2].

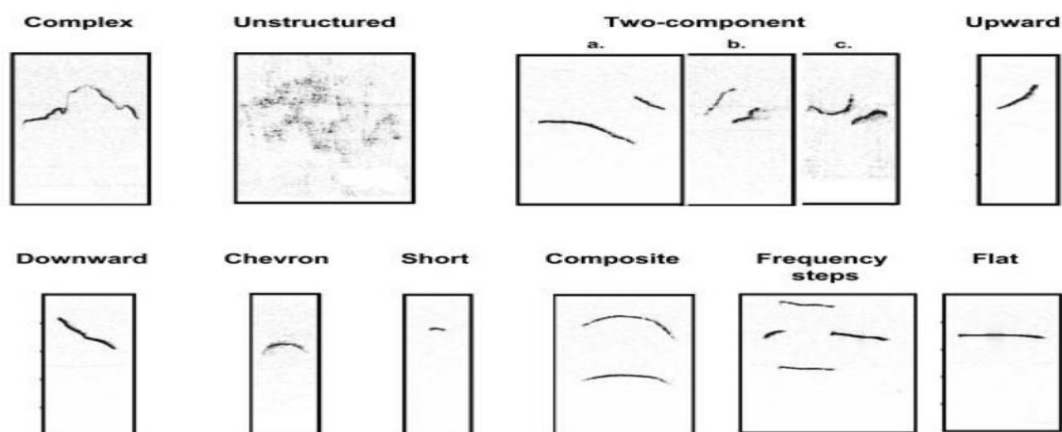


Figure 1: Typical sonograms of ultrasonic vocalizations, classified into ten distinct categories of calls emitted by adult mice [6]

Goals

To date there is no objective tool for the diagnosis of autism at young age. In fact, diagnosis is currently based on clinical inspection, including interview with the child parents, assessment of the child behavior by playing and contacting the environment. This is a relatively long process. Autism currently is not diagnosed before the age of 3 [1]. Early diagnosis can lead to effective treatment, which may significantly improve the child's functioning. The most common treatment nowadays is to provide positive

reinforcement for basic successful action of the child and negative reinforcement on the contrary. This treatment can be achieved at early ages, before the age of 3, based on the diagnosis.

Therefore, a new and early-age diagnosing tool is required. The aim of the project is to develop a sub-algorithm for diagnosing autism by vocalizations. This algorithm classifies mice's vocalic signals sounds and therefore allows to analyze and characterize the vocalic signals. The analyzing can give statistical information on autistic behavior and distinguish between a healthy mouse and a mouse with autistic behavior. If successful, the results of this study will be used for early identification of autism in children.

Hence, in order to diagnose one aspect of autistic-like behavior in mice, the project was focused on classification of USV recordings. The USV are usually expressed by spectrograms representation, based on which the syllables are estimated. There are 10 main syllables known in the literature that are identified by their pitch changes, length and shapes (Figure 1). The segmentation was made by an automatic rule-based algorithm implemented in the "Avisoft" software [7], while our previously developed segmentation algorithm was based on spectral energy distribution. There are small changes in the syllables discussed in each paper, but they look quite similar.

Analyzed the dataset at hand revealed that the syllables in "real life" may be a bit different than the syllables documented in the literature, but they keep their basic structure. Certain syllables, e.g., short, complex, chevron and flat, are very similar, making it difficult to label and classify them. One of the options to deal with the problem is to unify the similar syllables and thus reduce the types of syllables.

In Figure 2 we presented 10 samples for each syllable type we have in our dataset. Their classification was chosen by visualization of the sonograms from Figure 1 and by definitions of each syllable we show below. . For simplicity we represented the frequencies in the sonograms by colors, especially by the yellow color. The yellow color represents the frequencies with high amplitude in the spectrogram. Usually the strong yellow color represents the syllables especially if the components in the sonogram has a shape like in Figure 1. Otherwise, it can be noise.

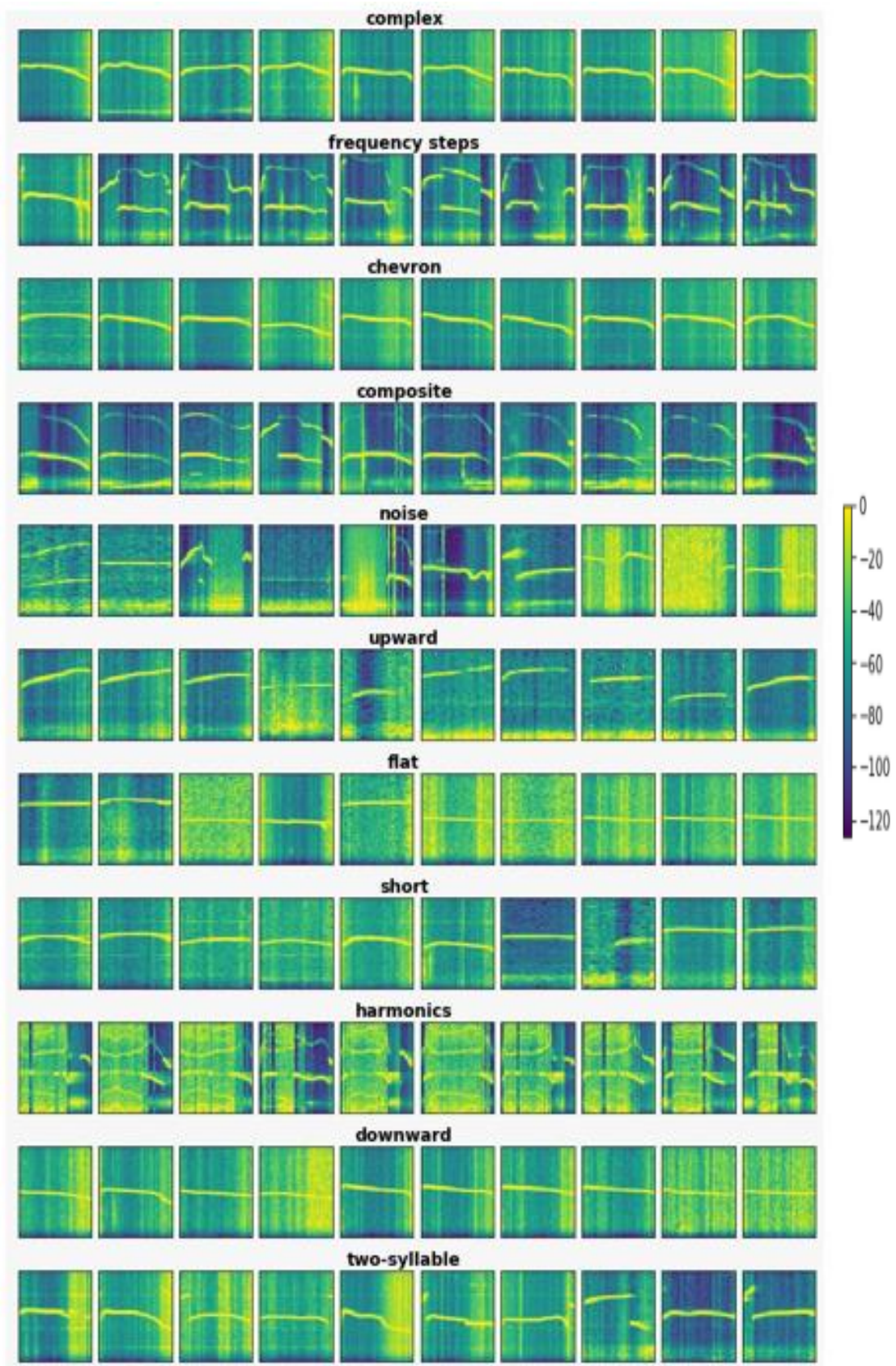


Figure 2: presentation of the 10 syllables of mice and noise with 10 examples each

Complex: One main syllable in the spectrogram that is continuous and has one or more Inflection points, usually spreads over the entire X axis (time)

Frequency steps: 3 or 4 components in the spectrogram, long one at low frequencies that spreads over almost the entire X axis, two short syllables, one at the beginning of the X axis and one at the end, both have higher frequencies than the component described above. If there are 4 syllables then another component is with higher frequencies than the first 3 and spreads similar to the first component described above, sometimes the last three components are almost connected.

Chevron: One main syllable in the spectrogram that is continuous and has a concave shape, sometimes it is hard to differ it from complex, usually spreads over the entire X axis

Composite: two main syllables in the spectrogram, one is at the medium range of the frequencies and the other has high frequencies, both spreads almost over the entire X.

Noise: everything that don't match to one of the other syllables, can be noisy signal without syllables, unstructured spectrogram, two or more syllable types in the same spectrogram for example frequency steps and another frequency steps inside the same figure.

Upward: one main syllable that has upward diagonal orientation from left to right.

Flat: one main syllable, that obtains totally flat orientation, meaning one main frequency.

Short: one main syllable, that is only at the middle of the X axis, this one is very problematic because we resize each spectrogram to constant size and thus it looks like complex or chevron and one option is to unite them, because we get the syllable length in seconds from the MATLAB, another option is to set a threshold for the syllable length so that under this threshold we will consider it short, we did not get to do it because we worked on different syllables.

Harmonics: ranges from three to seven components that spreads over the entire frequency range, the component with the intermediate frequency usually spreads over the entire X axis, while the components with higher and lower frequency usually shorter

in time, each component is usually with a flat orientation, sometimes can look like noise.

Downward: one main syllable that has downward diagonal orientation from left to right.

Two-syllable: two main syllables in the spectrogram, which be either long, short, may look very similar to frequency steps just with two components instead of three, and can also vary in the frequency range.

Our data comprise of recordings of USV emitted by mice at different ages. Recording of wild type and Mthfr-knock-out mice were included. The current dataset consists of 5891 (1953 of adults and 3938 of young) tagged syllables. We used a previously-developed model for segmentation . In Table 1 on the left side there is a distribution of the old data and on the right side there is a distribution of the updated data, arranged in descending order.

Table 1: distribution of classified USV's. On the left – the old data (adults). On the right – the new data (young)

Old data – adult mice			New data – young mice		
Syllable	Count	%	Syllable	Count	%
Frequency Steps	11	0.6	Frequency Steps	1451	36.8
Noise	2	0.1	Noise	611	15.5
Composite	71	3.7	Composite	573	14.6
Two-syllable	37	1.9	Two-syllable	430	10.9
Chevron	527	27.3	Chevron	333	8.5
Complex	579	30	Complex	226	5.7
Harmonics	14	0.7	Harmonics	147	3.7
Flat	164	8.5	Flat	75	1.9
Upward	322	16.7	Upward	72	1.8
Short	143	7.4	Short	13	0.3
Downward	83	4.3	Downward	7	0.2

The most interesting thing in Table 1 is the distribution difference between the data types – adult vs young mice: the adults use the complex syllable the most while the young use the frequency steps syllable. This conclusion is relevant for most of the

syllables. The noise signal was inserted to the data as a "syllable" because we needed to add new category for signals which cannot represent syllables. Figure 3 represents the distribution of the united data – adult mice with young mice.

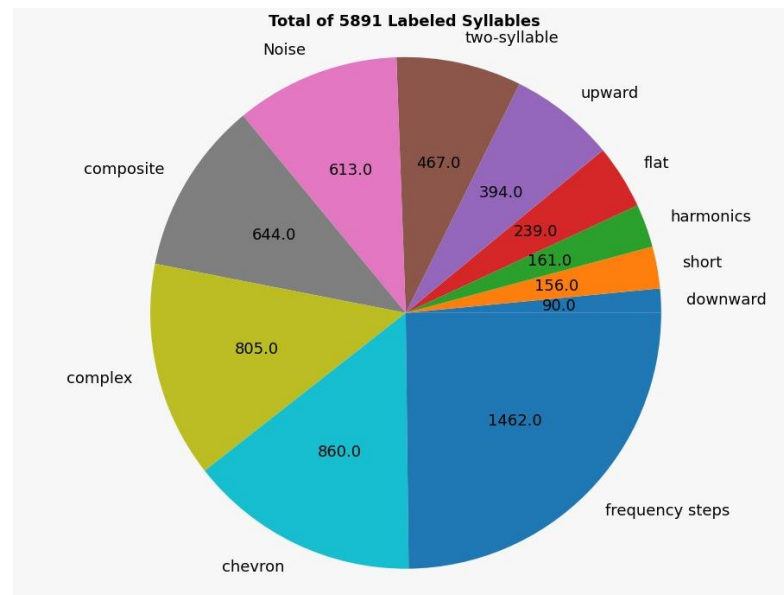


Figure 3: the quantities of four labeled syllables

Machine learning and deep learning

Despite strong evidence that USV serve an array of communicative functions, technical and financial limitations have been barriers for most laboratories to adopt vocalization analysis [3]. Manually processing of the vocalization data is time-consuming and subjective. We therefore decided to develop an automated machine learning approach for the analysis and processing of the USV signals.

Machine learning, a subset of artificial intelligence, is the study of computer algorithms that improve through experience. Those algorithms build a mathematical model based on a training data in order to make predictions or decisions about new data that have not been seen before [7].

Machine learning is divided into 3 main areas – supervised, unsupervised and reinforcement learning: supervised learning is a learning done through tagged data for

groups while unsupervised learning is a learning done without tagged data [7] and reinforcement learning is a behavioral learning.

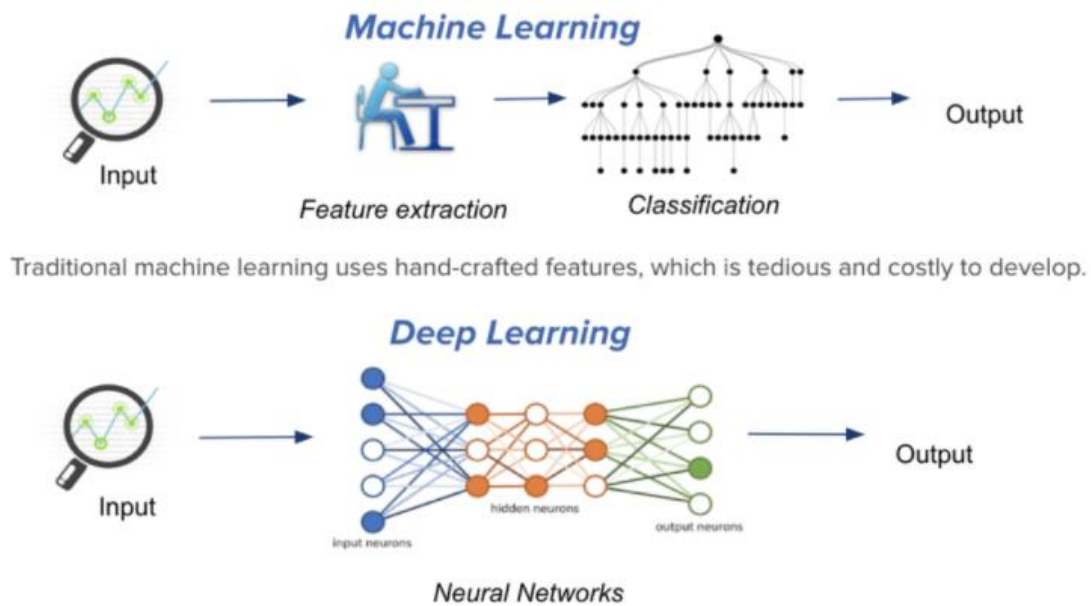


Figure 4: abstract scheme of machine learning vs deep learning [8]

Deep Learning – a family of methods within machine learning that uses available data to learn a hierarchy of representations useful for certain tasks. While in traditional machine learning a lot of human expert effort is needed to define the set of features to represent the data, there is no feature engineering involved in deep learning. The system learns the best representation of the data by itself to produce the most accurate results (Figure 4) [8].

Our innovation is to build an algorithm in an approach of unsupervised learning. We chose this approach because most of the data we have is not tagged. Tagging is an action that takes a lot of time and resources that we didn't have a lot from it.

In fact, most of the data is constitutes the training set of the model. This input data goes through a neural network (analogous to weights) and thus the model learns the relevant properties from each example. The output depends on the model type and can be an image, a number and so on [9].

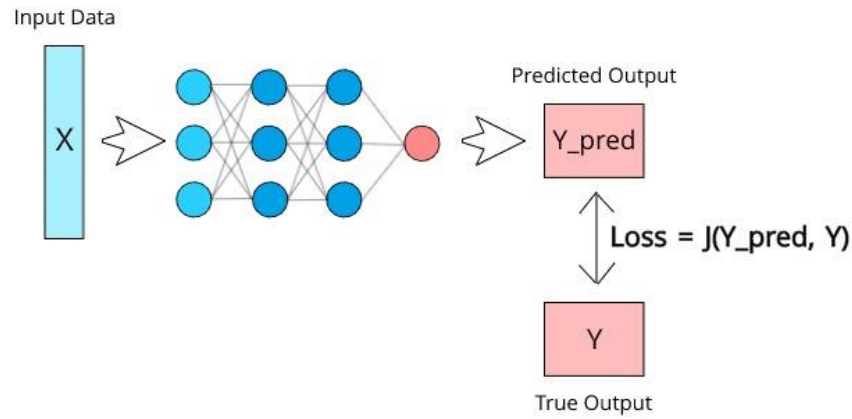


Figure 5: scheme of DNN model with loss function

In each model, a loss function must be defined. The model then minimizes the loss function, using back propagation and gradient descent to find the optimal weights (Figure 5).

Chapter 1 – Unsupervised learning

1.1 Methods

Autoencoder (AE) is an unsupervised artificial neural network that learns how to efficiently compress and encode data. It learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible [10]. AE, by design, reduces data dimensions by learning how to ignore the noise of the data. It consists two major parts – encoder and decoder [11].

The encoder generally uses a series of dense or convolutional layers to encode an image into a fixed length vector that represents the image as a compact form, while the decoder uses dense or convolutional layers to convert the latent representation vector back into that same image or another modified image [12].

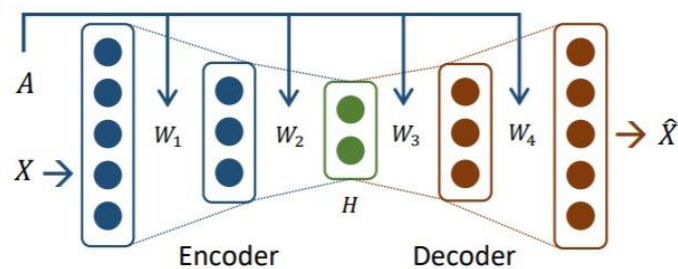


Figure 6: scheme of autoencoder [13]

In Figure. 6, the input of size X is compressed into a parameters vector of size Z and then decompressed into the same image of size X . To generate an image, a random input vector is given to the decoder network. The Decoder network will convert the input vector into a full image. This parameters vector represents the input by its parameters and with this low dimension vector we could use K-means for clustering the data.

K-means clustering is a common method for unsupervised learning. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K . The algorithm works iteratively to assign each data point to one of K groups based on the features. Data points are clustered based on feature similarity (Figure 7) [14].

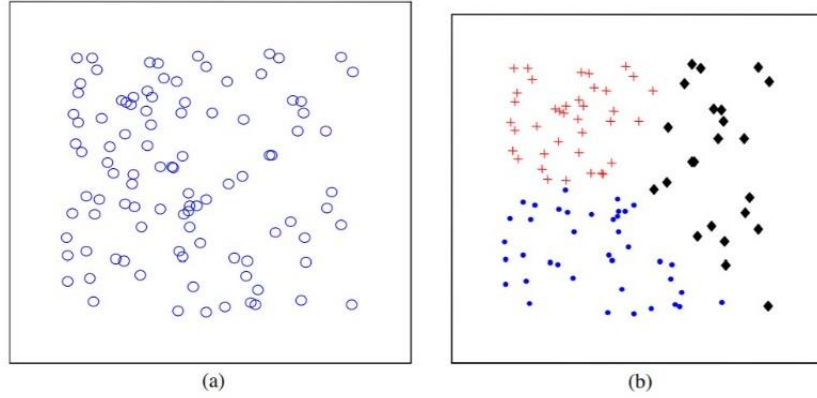


Figure 7: scheme of the K-Means method

1.2 Preliminary investigation using the MNIST dataset

First, as a proof-of-concept, we performed a preliminary investigation using the MNIST dataset. The MNIST is a large database of handwritten digits that is commonly used for training various image processing systems.

While using the data, we tried the autoencoder model with two syllables. We compared between the reconstructed syllables and the original syllables. After that, we used K-means method for the parameters vector and tested the results by confusion matrix to check the success of the model to classify right. This can happen only by sing known and classified data.

The performance of deep learning neural networks often improves with the amount of data available. Therefore, and due to the fact that our database is relatively small, we expanded it using augmentations. Data augmentation is a technique to artificially create

new training samples based on the existing ones. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples. For images, there is a lot of manipulations to do on them – horizontal and vertical shifting and flipping, random rotation, random brightness and so on [15]. We used small shifts vertically and horizontally because this shifting keeps the morphological shape of syllables’ spectrogram.

1.3 Results

Figure 8 presents the autoencoder model loss and reconstruction for 2000 samples.

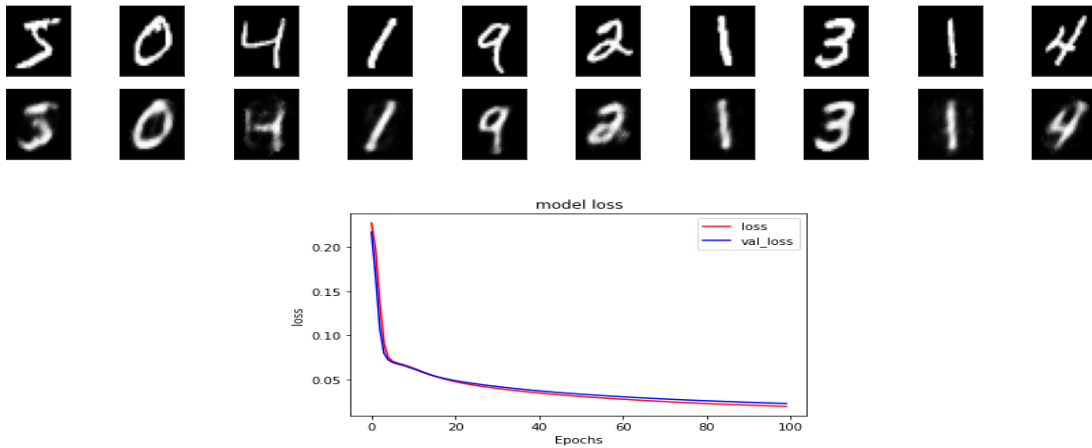


Figure 8: Loss and reconstruction of autoencoder model for 2000 samples from MNIST

Figure 9 presents the autoencoder model loss and reconstruction after augmenting the MNIST samples from 2000 to 10000 by slightly moving the images upwards, downwards and to the sides.

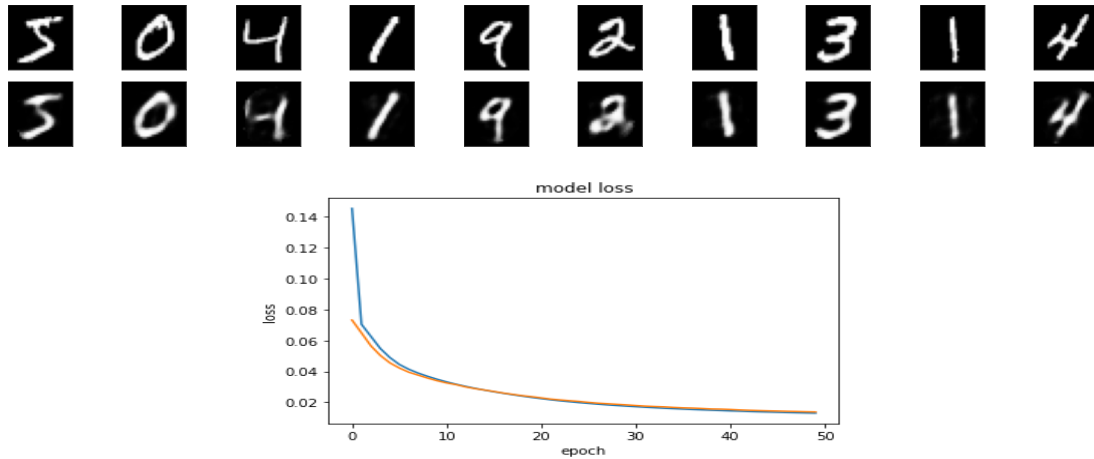


Figure 9: Loss and reconstruction of autoencoder model after augmenting from 2000 to 10000 samples from MNIST

Figure 10 presents the autoencoder model loss and reconstruction for 10000 samples.

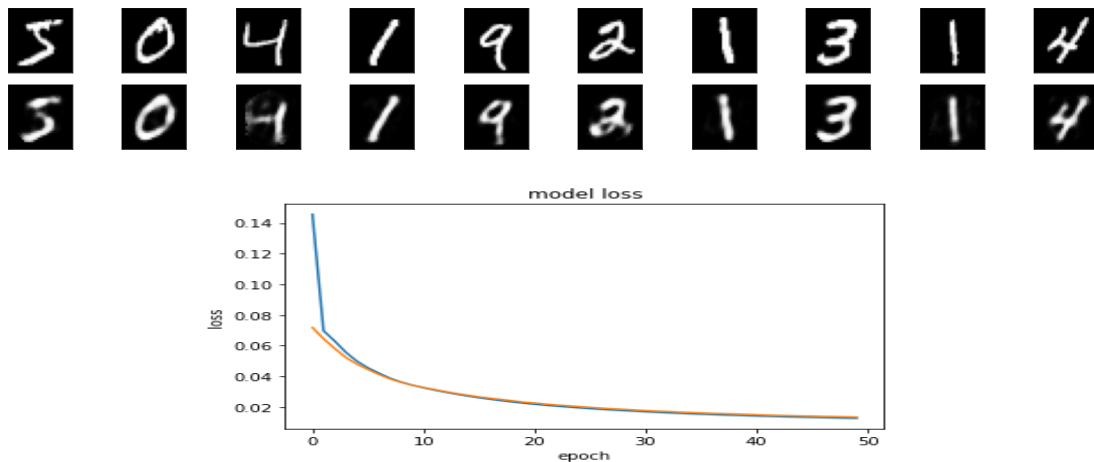


Figure 10: Loss and reconstruction of autoencoder model for 10000 samples from MNIST

Figure 11 presents reconstruction, loss, PCA and confusion matrix of the autoencoder model results on 1000 samples of 2 categories from the MNIST dataset – 6 and 3.

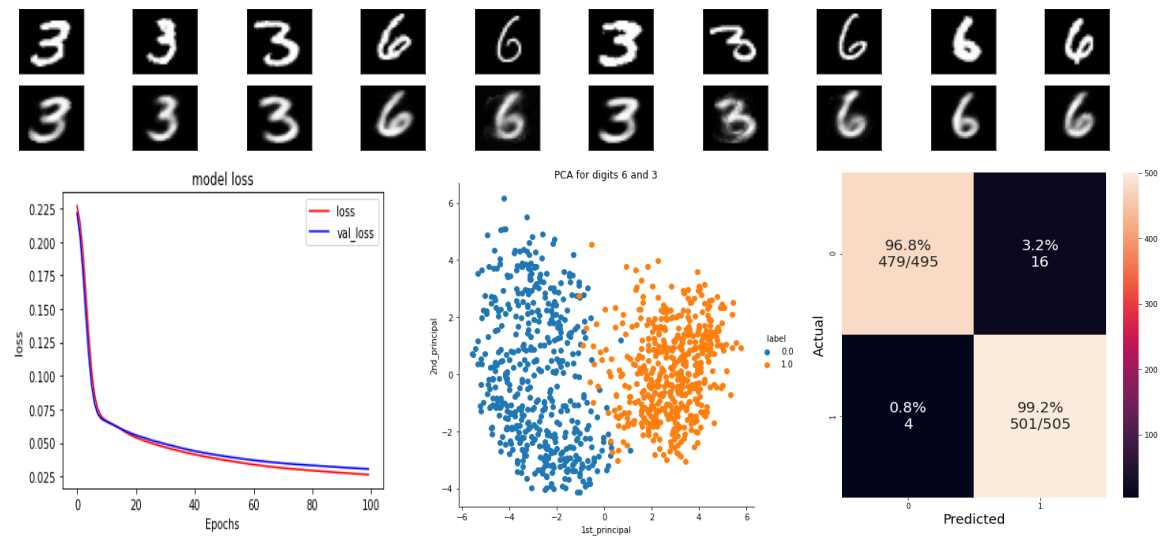


Figure 11: Images of reconstruction, loss, PCA and confusion matrix of the AE model on 1000 MNIST samples (6,3)

Figure 12 presents reconstruction, loss, PCA and confusion matrix of the autoencoder model results after augmenting from 1000 to 5000 samples 2 categories from the MNIST dataset – 6 and 3.

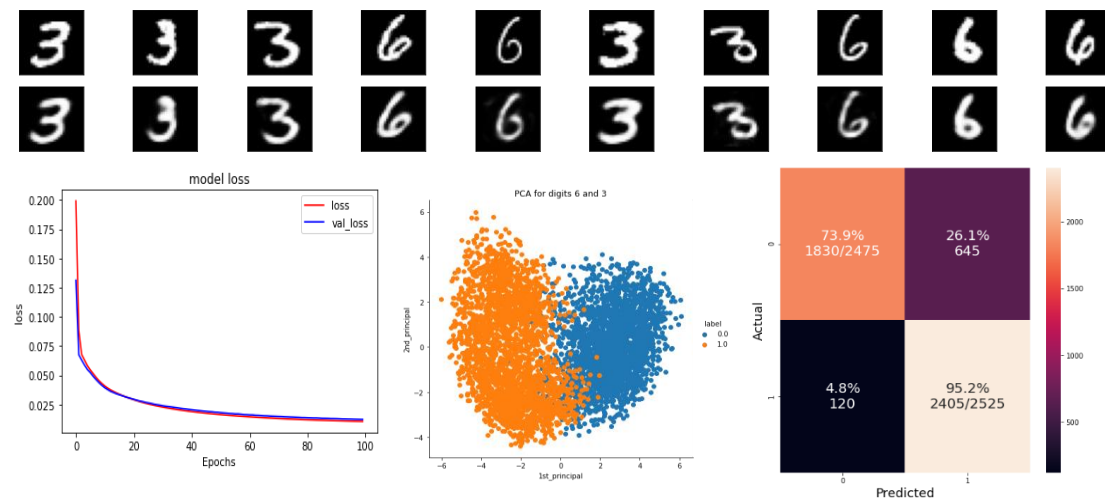


Figure 12: Images of reconstruction, loss, PCA and confusion matrix of the AE model on 5000 MNIST samples after augmentation from 1000 samples (6,3).

1.4 Classification of the USV dataset

Figure 13 presents reconstruction, loss, PCA and confusion matrix of the autoencoder model results. Images of reconstruction, loss, PCA and confusion matrix of the AE model on 1400 USV samples from 2 categories - Complex vs frequency syllables. The first row of spectrograms is the original signal spectrogram while the second row is

the reconstructed spectrogram, as we can see, the model reconstruct almost all of the spectrograms to one single form that tend to look as frequency steps.

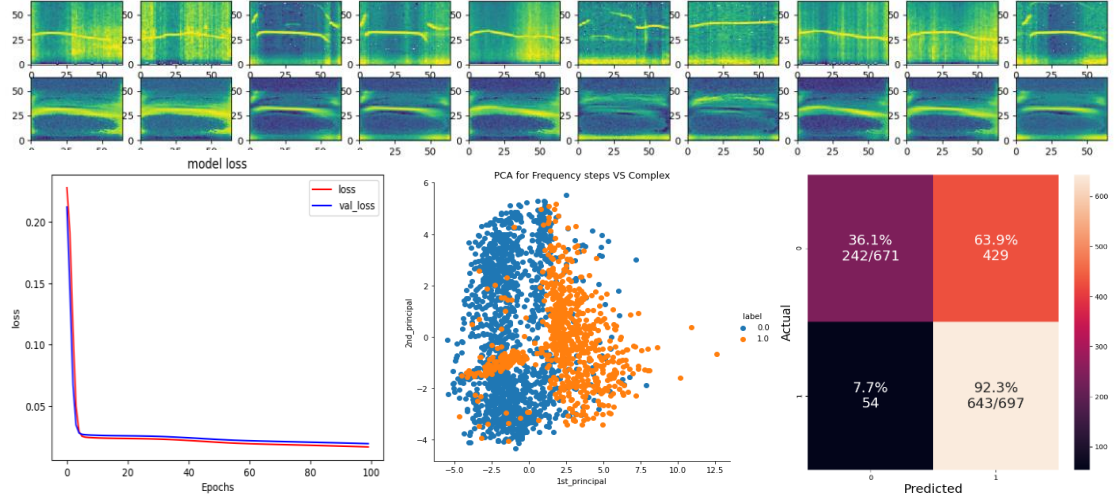


Figure 13: Images of reconstruction, loss, PCA and confusion matrix of the AE model on 1400 USV samples from 2 categories - Complex vs frequency syllables

1.5 Discussion

We performed a proof of concept using AE models on MNIST. Firstly, we just demonstrated the reconstruction and loss of the model on 2000 MNIST samples, then we used augmentations to increase the dataset in order to achieve better results because we know that DNN models works best with big data, the reconstruction is indeed better (visually) and the loss is lower. We saw that the reconstruction results of the model on 10000 original samples were very similar to 2000 samples with augmentations to 10000 (visually).

Next, we demonstrate the reconstruction, loss, confusion matrix and PCA after using the AE model on a total of 1000 samples of the digits 6 and 3 in the MNIST dataset. The model separation accuracy is very high – 99% and 96% of success in each category in the confusion matrix, the reconstruction is also quite clear (visually), the loss converges and the PCA shows quite clear separation.

We then used augmentations to enlarge the dataset from 1000 samples to 5000 and surprisingly we got worse results, even though the reconstruction looks better (visually) and the loss is the same as before the augmentations, the K-MEANS clustering accuracy is only 74% and 95% in the confusion matrix, and the PCA clusters overlap each other. We assume that this difference is due to the fact that the augmentations enlarge the variance and because of the shifting of the images the algorithm find it difficult to see

different structures for the two digits, the MNIST data is centered meaning the digits always in middle of the figure and once we shifted the images from the center the model performs worse.

Our data is USV syllables which is significantly different (visually) than the MNIST dataset. We achieved poor results on our dataset - the reconstruction tend to have one major form of the frequency steps even when it is not the case, the loss converges but the K-MEANS clustering accuracy is 36% of one syllable category and 92% of the second, it predicts more than 1000 as one category and less than 300 for the second category while the true split is about 50% – 50%.

Comparing to previous projects in Golan group, the unsupervised learning method has a lot of potential due to fact that almost all of the mice USV is not labeled, and a good clustering of the syllables would benefit a lot in analyzing the differences between healthy mice and mice suspected with ASD. Unfortunately, the results are not satisfying so we went back to the supervised learning approach.

Chapter 2 –Supervised learning

2.1 Methods

Another approach we used is classification using Convolutional neural network (CNN) method. CNN is one of the main architectures in the field of computer vision, for instance in objects detection and face recognition tasks. CNN image classifications take an input image, process it and classify it under certain categories.

Technically, deep CNN models able to train and test data. Each input image passes through a series of convolution layers with filters, pooling, fully connected layers (FC) to classify an object. The Figure below is a general scheme of CNN to process an input image and classifies the objects based on values [16].

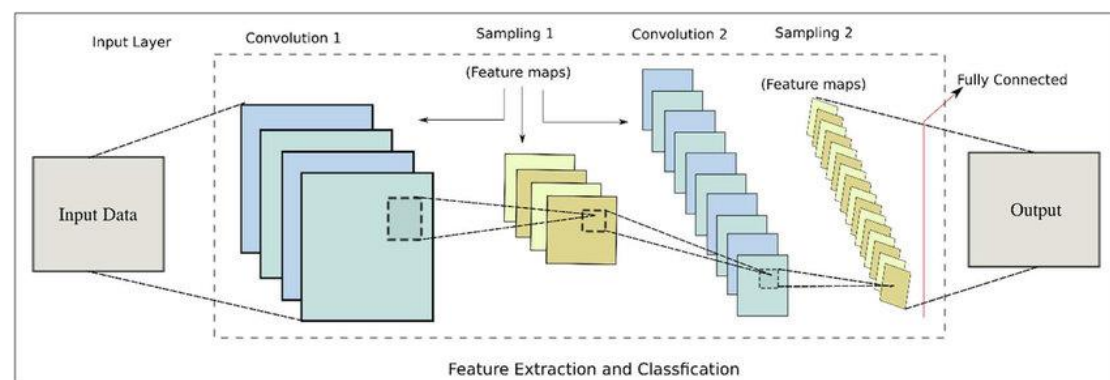


Figure 14: CNN model [17]

In CNN, the data is divided into three sets – training, validation and test [18]:

- **Training set** – the portion of dataset used to fit the model.
- **Validation set** – the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters.
- **Test set** – the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

In our project, we used the model architecture shown in Figure 15.

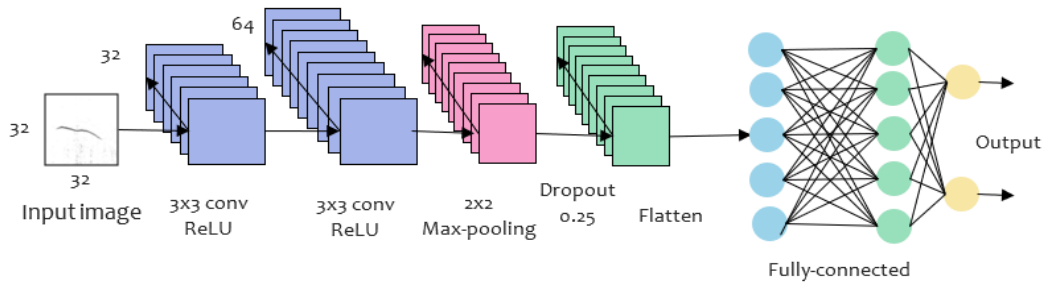


Figure 15: The CNN model we used – contains 2 convolutional layers with ReLU activation function, a max-pooling layer, a drop-out layer, a flatten layer, and fully connected layers [19].

The model consists several layers [20]:

- **Convolution layer + ReLU** – a “filter” passes over the image, scanning a few pixels at a time and creating a feature map that predicts the class to which each feature belongs. Non-Linearity (ReLU) stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0, x)$.
- **Max pooling** – Pooling layers section would reduce the number of parameters when the images are too large. Max pooling takes the largest element from the rectified feature map.
- **Flatten** – flattening the feature map matrix into a vector in order to feed it into a fully connected layer.
- **Fully connected layers** – the first layer takes the inputs from the feature analysis and applies weights to predict the correct label through the other layers. The output layer gives the final probabilities for each label.

Deep learning neural networks are trained using the stochastic gradient descent optimization algorithm. As part of the optimization algorithm, the error for the current state of the model must be estimated repeatedly. This requires the choice of an error function, conventionally called a loss function, that can be used to estimate the loss of the model so that the weights can be updated to reduce the loss on the next evaluation [9]. We performed several experiments using CNNs.

Our data comprise of recordings of USV emitted by mice at different ages. Recording of wild type and Mthfr-knock-out mice were included. The current dataset consists of 5891 (1953 of adults and 3938 of young) tagged syllables. We used a previously-developed model for segmentation . In Table 1 on the left side there is a distribution of the old data and on the right side there is a distribution of the updated data, arranged in descending order.

2.2 Results

Figure 16 presents loss and accuracy of the model for a total of 1600 samples from two categories - Complex vs frequency syllables.

The model training and validation loss converge quite fast (after less than 15 epochs). We can see that the model begins to overfit after the eighth epoch (we take the model that performed best using model checkpoint callbacks), the train and validation accuracy are high – above 95% which is quite satisfying.

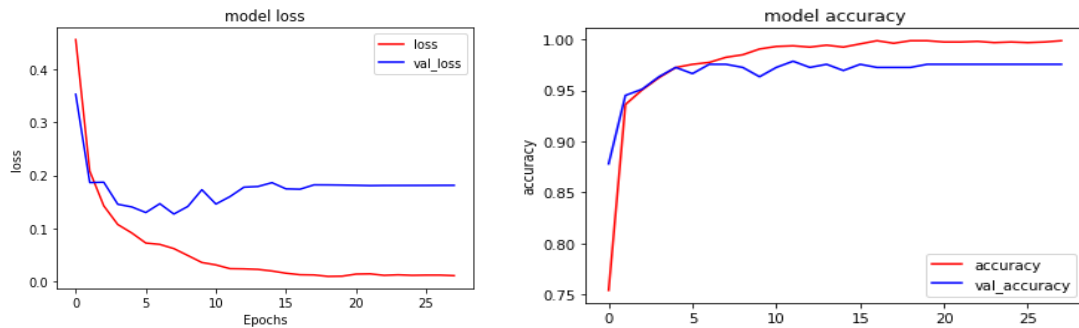


Figure 16: loss, and accuracy for 2 categories - Complex vs frequency syllables

Figure 17 presents examples of the model errors and the probability of each prediction for total of 1600 samples from 2 categories - Complex vs frequency syllables. By visually examine the errors we can see what syllables the model misclassified and try to

understand what caused the mistakes, for example images 2,3 indeed look like complex but were misclassified.

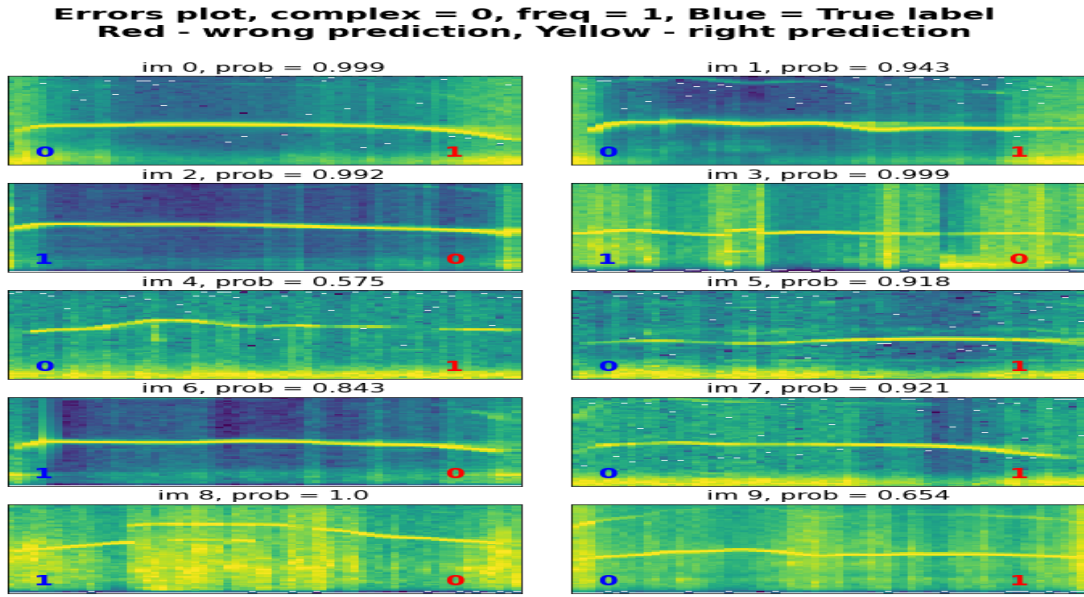


Figure 17: examples of the model errors and the probability of each prediction

Figure 18 presents general examples of the model predictions and the probability of each prediction for total of 1600 samples from 2 categories - Complex vs frequency syllables.

In addition, it can be noticed that the probability for the correct predictions seems to be higher comparing to the probability of the error predictions.

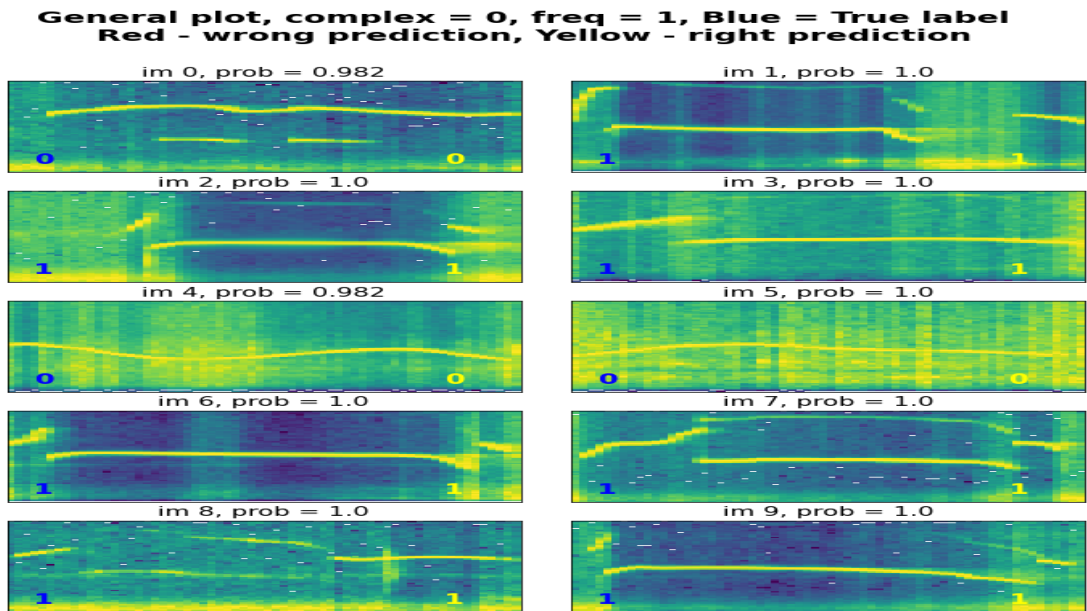


Figure 18: examples of general predictions and the probability of each prediction

Figure 19 presents loss and accuracy of the model predictions after augmentations for total of 1600 samples from 2 categories - Complex vs frequency syllables.

Our dataset is relatively small and therefore we tried augmentation to enlarge the dataset. The augmentations were a mild shifting of the spectrograms to the sides, upward and downward. The train and validation loss and accuracy are quite similar to the results before the augmentations.

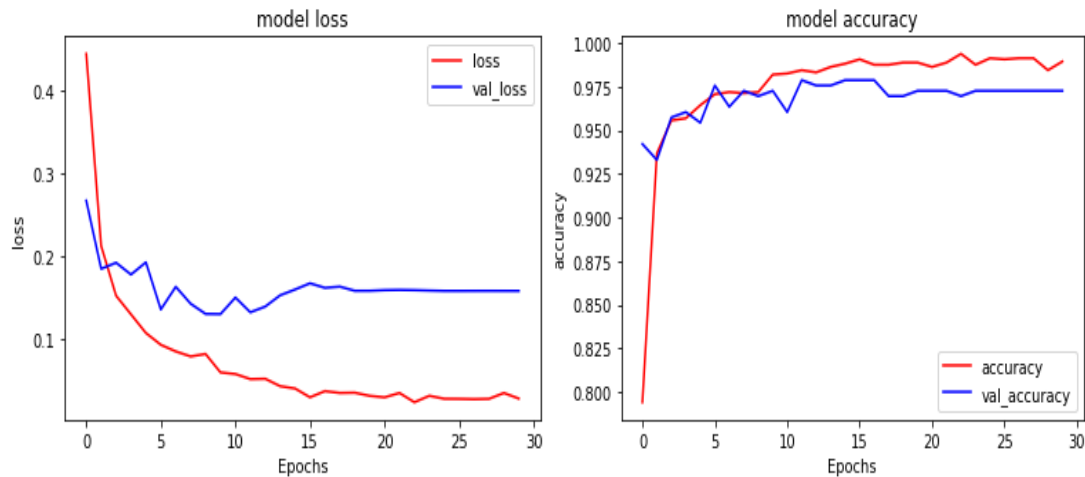


Figure 19: loss and accuracy of the model predictions for 2 categories after augmentation - Complex vs frequency syllables

Table 2 presents the confusion matrix of the model for total of 1600 samples from 2 categories - Complex vs frequency syllables.

Table 2: confusion matrix of the model predictions for total of 1600 samples from 2 categories- Complex vs frequency syllables.

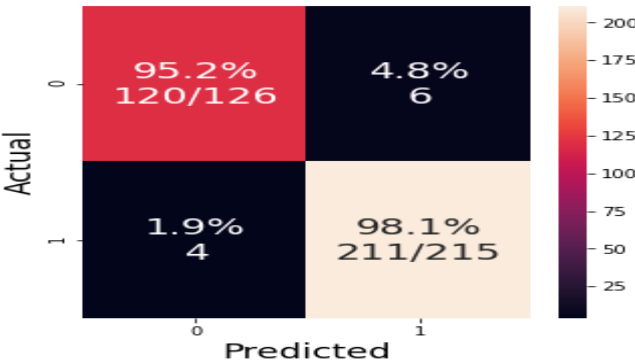


Table 3 presents the confusion matrix of the model predictions after augmentations for total of 1600 samples from 2 categories - Complex vs frequency syllables.

The results on the test set did not change dramatically before and after the augmentations – the model misclassified 14 comparing to 12 before (Figure 17), thus we can assume that even the mild shifting was wrong and caused for the loss of important information.

Table3 : confusion matrix of the model predictions after augmentations for total of 1600 samples from 2 categories - Complex vs frequency syllables

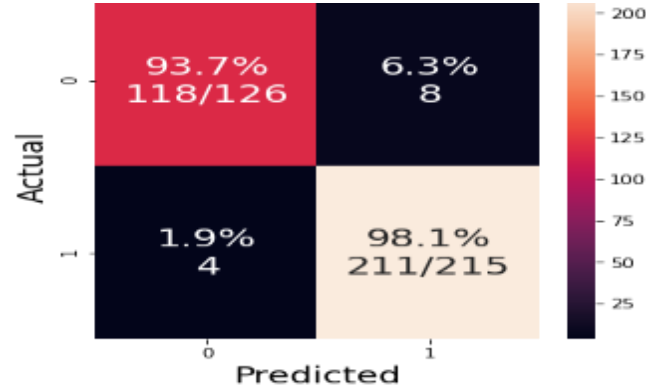
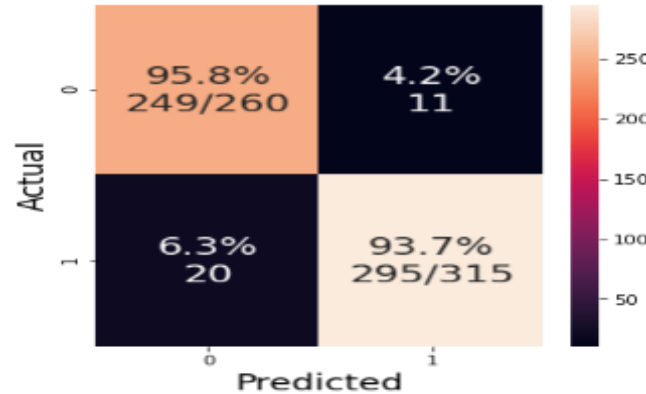


Table 4 presents the confusion matrix of the model for a total of 2700 samples from two categories – united Complex, chevron and flat vs frequency and two-syllable syllables.

Because that flat, chevron and complex syllables can look similar, and two-syllable and

Table4 : confusion matrix of the model for total of 2700 samples from 2 categories – united Complex, chevron and flat vs frequency and two-syllable syllables.



frequency steps can look similar as well, we performed an experiment that tries to reduce the number of categories and we combined the above syllables. We compared between the two different groups and achieved high accuracy results – the model misclassified 31 samples out of 575, meaning accuracy of 94.6%

Table 5 shows the confusion matrix of the model predictions for total of 2000 samples from 3 categories– Complex vs frequency vs composite syllables.

Next, we tested the model on 3 categories, we chose complex vs frequency vs composite syllables and the model misclassified 32 out of 447 test samples (92.8% accuracy),

which are still quite promising results, even though sometimes composite and frequency step syllables can look alike.

Table5 : confusion matrix of the model predictions for total of 2000 samples from 3 categories– Complex vs frequency vs composite syllables.

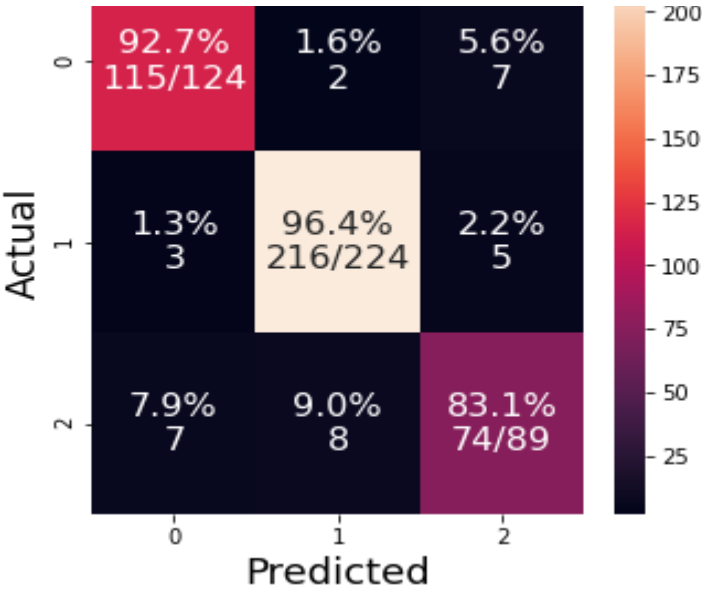
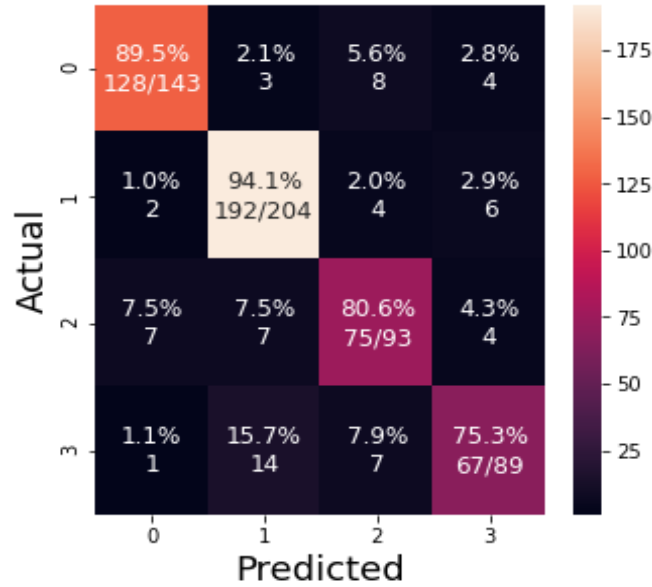


Table 6 presents the confusion matrix of the model predictions for total of 2500 samples from 4 categories – Complex vs frequency vs composite vs noise syllables.

The last experiment with CNN model was for four different categories: complex, frequency steps, composite and noisy syllables. The model misclassified 67 syllables out of 529, i.e., an accuracy of 87.3%.

Table6 : confusion matrix of the model predictions for total of 2500 samples from 4 categories – Complex vs frequency vs composite vs noise syllables



2.3 Discussion

As the results show, supervised learning is an easier task and achieves much more accurate results comparing to the unsupervised learning method.

The results are quite satisfying even for the different categories. Combining similar syllables under the same label can be useful for future analysis in order to reduce the number of categories and therefore reduce the complexity of the problem.

The model training loss and the validation loss converge quite fast (after less than 15 epochs in all the experiments above).

The augmentations did not improve the results, apparently due to the fact that important information was lost. Different augmentations method should be considered.

It is important to visualize the errors of the model because mistakes in labeling can easily happen. Additionally, low probability prediction can point on syllables which the model finds hard to classify.

In comparison to previous projects in Golan group, we upgraded the CNN model to be more general and to achieve better results. Additionally, we convert the python preprocessing of the audio signals to be almost the same as the MATLAB preprocessing

from previous project, (with addition of reshaping the spectrograms to 64*64 for the CNN model), we made a very user friendly code on google Colab for the unsupervised and supervised model, we noticed a very big problem of diversity in the labeling of the syllables so we wrote a guide for how to label in a uniform way, and we increased the labeled data amount by more than 3000 samples.

Chapter 3: active learning

3.1 Methods

Active Learning, as the name implies, is a method used to increase the number of tagged samples in a dataset, based on user feedback. The algorithm chooses which results it keeps for tagging manually in order to improve a model by taking specific results. This choice is made by specific method that takes samples with probability prediction under specific percentage [21].

Active learning has three different settings to work with. Figure 9 shows a general scheme of the pool-based sampling setting of active learning method used in this work. This setting assumes that there is a large pool of unlabeled data. Instances are then drawn from the pool according to some informativeness measure. This measure is applied to all instances in the pool and then the most informative instances are selected. This is the most common setting in the active learning community. The strategy we used for query instances is called Least Confidence (LC) [21].

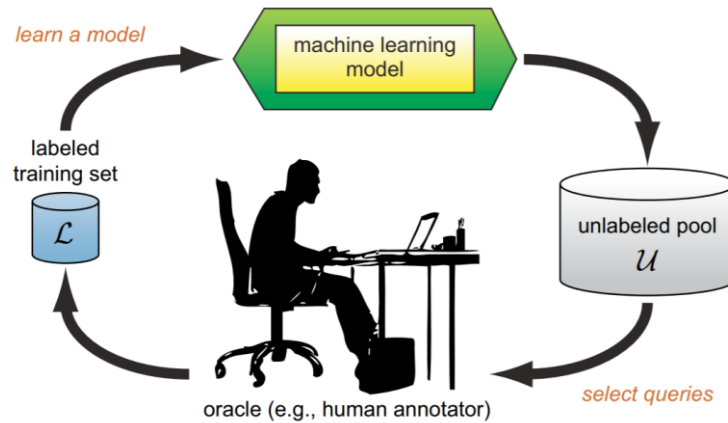


Figure 20: The pool-based active learning cycle [21]

We used our trained model and tested a set of untagged syllables that the model classified according to probabilities. The probabilities can show us the degree of confidence of the model in choosing the right class. Then, we took all the 33 syllables

whose probability to be correct is lower than a certain threshold (in our case – 80%). We classified those syllables manually by tagging them and then used the new set we received, which includes the syllables the model classified with the low probability syllables. We would anticipate getting better classification than if we would add the original test set without tagging the syllables manually as described above.

3.2 Results

For the AL method, we used the CNN model on two types of syllables – Complex and Frequency Steps (FS) marked as 0 and 1, respectively. The data contained 600 examples in the train set (300 Complex and 300 FS), 200 examples in the test set (96 Complex and 104 FS) and 100 examples in the validation set. This test set is called test set A. The other test set we used is called test set B. Test set B is the unseen and unlabeled set on which we can activate the AL algorithm.

To determine the AL efficacy, the model first predicted the classifications for test set B. The results are summarized in Figure 21. The model succeeded to classify the Complex syllable correctly by 81.2% accuracy and 85.6% accuracy for the FS syllable. Also, we can see in the loss graph that the model is converged and the accuracy too.

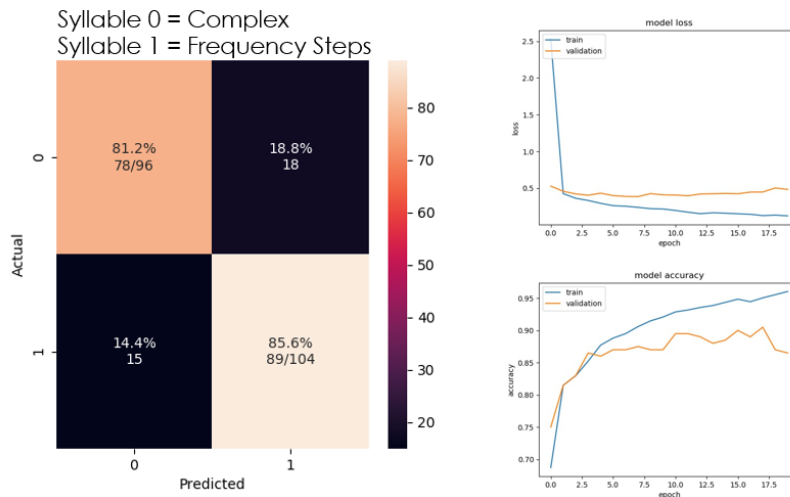


Figure 21: loss and accuracy of the CNN model (right) and the confusion matrix of test set A (left)

After the model predicted the classifications, we took only the syllables with probabilities lower than 0.8 and classified them based on visual inspection. The new test set A was added to the train set. After re-training the model, it predicted the classification for test set B. In Table 7 the confusion matrix on the right is presenting the results of the classifier – 84.4% for Complex and 83.7% for FS.

In contrast, we tried training the model with the regular train set plus test set A without using AL. The results on test set B are shown in Table 7 on the left. The classifier predicted correctly 80.2% for Complex syllable and 83.7% for FS syllable.

Table 7: Confusion matrix of test set A after using AL on test B (right) and confusion matrix of test set A without using AL

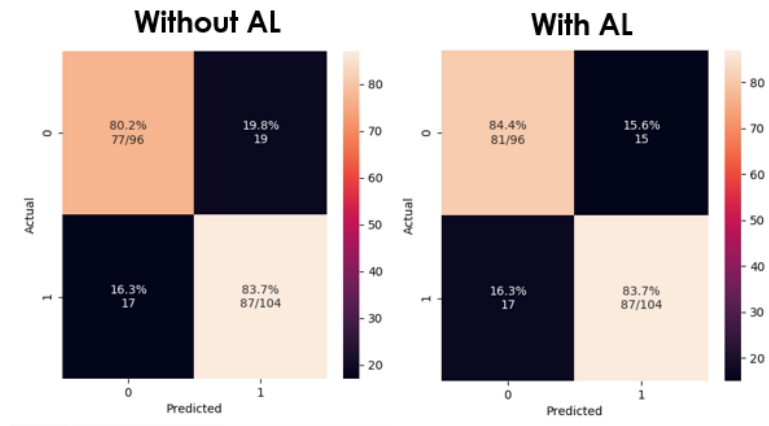


Table 8: the correctly predictions for test set B in the 3 different checks

Syllable/train set	Original train set	Original train set + unchanged test set A (without AL)	Original train set + changed test set A (after AL)
Complex	81.2%	80.2%	84.4%
Frequency Steps	85.6%	83.7%	83.7%

3.3 Discussion

As we can see in Table 8, without using AL, the prediction success of the Complex syllable decreased by 1% and for using AL it increased by 3.2%. For the FS syllable, there was no difference between using AL or not – the prediction success was 83.7% for both – a decrease of 1.9%.

We can conclude that test set B did not improve the FS syllable classification performance but by using the AL method, the classification of the Complex syllable was better than before using. We expected to get better results after using the active learning method because we "helped" the model by giving it classes for syllables that it had hard to know in high probability. Nevertheless, the percentages of classification were still satisfactorily improved.

This field of AL is pretty new from the last years, so there is not a lot of studies which used this method in order to get perspective on the results we got. We believe that in the next decade the use of it will increase and the confident of realize AL will be clearer.

Summary

In this project, we have investigated the potential of using deep neural networks, for the classification of USV to different categories. Various unsupervised learning models were evaluated with poor accuracy results of 64% for only two syllable categories, while using supervised learning with CNN model benefited more than 95% for 2 categories, 90% for three categories and 85% for four categories. Thus, it appears that unsupervised learning is not suitable for our purpose and is a much harder mission than supervised learning. For the AL method, we got an improvement for the 2 checked classes – Complex and Frequency Steps.

In the future, we offer to combine similar looking syllables in order to make it easier for the labeling mission and for the model classification as well, thus solving the problem of lack of data for the minor categories. Additionally, it is very important to make sure that the labeling mission is being carried out under the same instructions and the labeling is uniform.

References

- [1] M. L. Scattoni, S. U. Gandhi, L. Ricceri, and J. N. Crawley, "Unusual Repertoire of Vocalizations in the BTBR T+tf/J Mouse Model of Autism," *PLoS One*, vol. 3, no. 8, pp. 48–52, 2008.
- [2] M. Wöhr and R. K. W. Schwarting, "Affective communication in rodents : ultrasonic vocalizations as a tool for research on emotion and motivation," *Cell Tissue Res.*, vol. 354, no. 1, pp. 81–97, 2013.
- [3] K. R. Coffey and R. G. Marx, "DeepSqueak : a deep learning-based system for detection and analysis of ultrasonic vocalizations," *Neuropsychopharmacology*, vol. 44, no. 5, pp. 859–868, 2019.
- [4] M. Luisa, J. Crawley, and L. Ricceri, "Ultrasonic vocalizations : A tool for behavioural phenotyping of mouse models of neurodevelopmental disorders," *Neurosci. Biobehav. Rev.*, vol. 33, no. 4, pp. 508–515, 2009.
- [5] S. E. Bryson, S. J. Rogers, and E. Fombonne, "Autism Spectrum Disorders: Early Detection, Intervention, Education, and Psychopharmacological Management," *Can. J. Psychiatry*, vol. 48, no. 8, pp. 506–516, 2003.
- [6] M. L. Scattoni, L. Ricceri, and J. N. Crawley, "Unusual repertoire of vocalizations in adult BTBR T+tf/J mice during three types of social encounters," *Genes, Brain Behav.*,

- vol. 10, no. 1, pp. 44–56, 2011.
- [7] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. MIT Press, 2014.
 - [8] A. Ciptadi, “Deep Learning vs Machine Learning,” *BLUE HEXAGON BLOG*, 2019. .
 - [9] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, “An introduction to deep learning,” in *ESANN 2011 proceedings, 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2010.
 - [10] E. Banijamali and A. Ghodsi, “Fast spectral clustering using autoencoders and landmarks,” *arXiv.org*, 2017.
 - [11] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, “auDeep: Unsupervised learning of representations from audio with deep recurrent neural networks,” *arXiv.org*, 2017.
 - [12] R. Bank, D., Koenigstein, N., Giryas, “Autoencoders,” *arXiv.org*, 2020.
 - [13] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, “Symmetric graph convolutional autoencoder for unsupervised graph representation learning,” *arXiv.org*, 2019.
 - [14] W. Kwedlo, “A clustering method combining differential evolution with the K-means algorithm,” *Pattern Recognit. Lett.*, vol. 32, no. 12, pp. 1613–1621, 2011.
 - [15] Y. Wang, X. Wei, X. Tang, H. Shen, and L. Ding, “CNN tracking based on data augmentation,” *Knowledge-Based Syst.*, vol. 194, 2020.
 - [16] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach,” *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.
 - [17] B. Jan *et al.*, “Deep learning in big data Analytics: A comparative study,” *Comput. Electr. Eng.*, vol. 75, pp. 275–287, 2019.
 - [18] A. Ng, “Machine Learning Yearning-Draft,” *Open Draft*, 2016.
 - [19] S. Tzur and R. Altshular, “Analysis of vocal signals in mice to identify Autistic behavior,” 2019.
 - [20] S. J. Lee, T. Chen, L. Yu, and C. H. Lai, “Image Classification Based on the Boost Convolutional Neural Network,” *IEEE Access*, vol. 6, pp. 12755–12768, 2018.
 - [21] B. Settles, “Active learning literature survey,” 2009.