Children's Toy

Ana-Maria Musca, Alara Karadeniz, Yasin Karyağdı

Introduction

The goal of this project is to create a prototype for a voice recognition system. In particular we want to train a classifier that can recognize a multitude of words spoken by adults and children alike. We want to use this prototype in order to create a children's toy which uses an actuator to respond to classified words with auditory responses. We found that there is an importance in auditory learning and therefore wish to improve the auditory learning experience such that we provide an educational and fun experience.

Literature Study

For this literature study we have found two articles to get ideas from.

We first looked at a study called *Automatic Classification of Anuran Sounds Using Convolutional Neural Networks*[1]. In this study it is talked about how using an MFCC is the most popular way to process sound for the purposes of classification. We initially had the idea to use FFTs to classify our data, but this study made us realize that using MFCCs might be a better option for our classifications. This would also help to make our project work with different voices as it helps to mimic human speech production and perception. Also looking at sections 3.1 and 3.2 we can get some ideas on the preprocessing of the sounds we will be feeding into our neural network, and we can also get some ideas on how to extract the MFFCs through it.

Then we looked at a study called *Speech Sound Classification and Estimation of Optimal Order of LPC Using Neural Network*[2]. Looking at this paper and seeing that they have used a VAD (Voice Activity Detector) has made us think about the possibility of using it. We were thinking of how necessary it may be and how much it may over complicate our project, which we realize may be the case and decided against it. The section 2.3 about neural networks was also an interesting read as it talked about different options, but we ultimately decided to use a regular neural network. In section 4 they talk about how they test out the program through speech samples found online, which could be a great way in our project to test how the program recognizes different voices.

Concept of Operation

Concept Idea

Children love animals and especially to read books in which their favorite animals can be visualized. But the only thing that this kind of book lacks is the sound imitation of those animals. Sound can bring the world of animals closer to the children and with this idea in our head we decided to create a system in the form of a toy which will teach the children about different animals and the sounds they create. As a result, the children could learn in a fun way about the

[1] https://dl.acm.org/doi/10.1145/2948992.2949016
[2] https://dl.acm.org/doi/10.1145/3271553.3271611

animals that they are interested in and also develop their intellect by attempting to teach themselves regarding this topic without the assistance of the parents. Moreover, with the help of our toy, skills such as speech and sound recognition in children can be improved by letting them interact with the toy by speaking to it.

Stakeholders

- The stakeholders are the parents as they invest in our toy system by buying it. It can be said that a reason for them to be interested in our toy is that it provides a fun and interactive way for their children to learn about the animals around them.
- We can also be considered stakeholders as we invested time and effort in creating this system.
- We can, also, consider the children as stakeholders as they are the main recipient of our system.

Scenarios

- Our system is an interactive and fun way to help children teach themselves more about the animals surrounding them. Since learning by watching documentaries or cartoons, interactive sound toys like the one created by us, help children to identify different animals by classifying them by the sound that a certain animal makes.
- This toy might also help blind children get a larger view about what the animal world is like as a lot of the toys made with this concept use a lot of visuals, but our design is purely based on sound.
- Our system can be used as a background noise generator for relaxation (like ocean or rain sounds) by simulating the sounds of a day at a farm or an expedition in the African savanna (feature depending on the preferences of the client)

System Requirements

Specifying the Requirements

Functional:

- The system shall have the capability to record audio.
- The system shall have the capability to process the recorded audio.
- The system shall have the capability to classify its input to one of the predetermined classes.
- The system shall have the capability to output audio.

Non Functional:

- The classifier shall have an accuracy above 90%.
- The system shall have the capability to classify the inputs regardless of the person.

- The product shall be appealing to children.
- The sound level of the actuator shall be adjustable.

Constraints:

- The system must be developed in Matlab.
- The system must be ready by 22/10/2023.

Prioritizing the Requirements

Due to limitations in terms of time and means we decided to prioritize our requirements with the MoSCoW method. For our project it is important that we can demonstrate our prototype, to this end we found that we must be able to give the computer an input, and the computer must be able to give an output. In order to give a correct output the system must contain a classifier which has been trained. Since we want to give the Mel-Frequency Cepstral Coefficients as inputs to the classifier we must process the audio input into Mel-Frequency Cepstral Coefficients. It would be great to have the output be in terms of audio but we also found that the output could also be printed to the console, which would save us some time. Additionally if possible we would like the classifier to have a high accuracy since this would be ideal, if however we fail to properly set up the classifier then a slightly lower accuracy would be acceptable. Lastly it is important to note that in theory the use of the MFCC should allow the classifier to classify the input correctly even in the case that the person giving the input was not included in the training set of the classifier. In the case that we fail to properly process and train the classifier with the MFCC this unfortunately will not occur. If such an event were to happen we would have to temporarily give up on this requirement, thus we might debate whether there would be any use in giving MFCC as an input and therefore might also consider switching to giving the frequency spectrum as an input. With all this in mind we made the following prioritization:

Must have:

- The system shall have the capability to record audio.
- The system shall have the capability to process the recorded audio.
- The system shall have the capability to classify its input to one of the predetermined classes.

Should have:

- The system shall have the capability to output audio.
- The classifier shall have an accuracy above 90%.
- The MFCC shall be given as an input to the classifier.
- The system shall have the capability to classify the inputs regardless of the person.

Could have:

- The product shall be appealing to children.

Won't have

- The sound level of the actuator shall be adjustable.

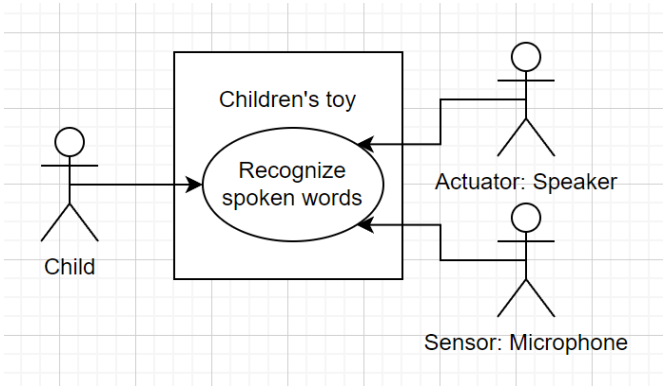


Diagram 1: User Case

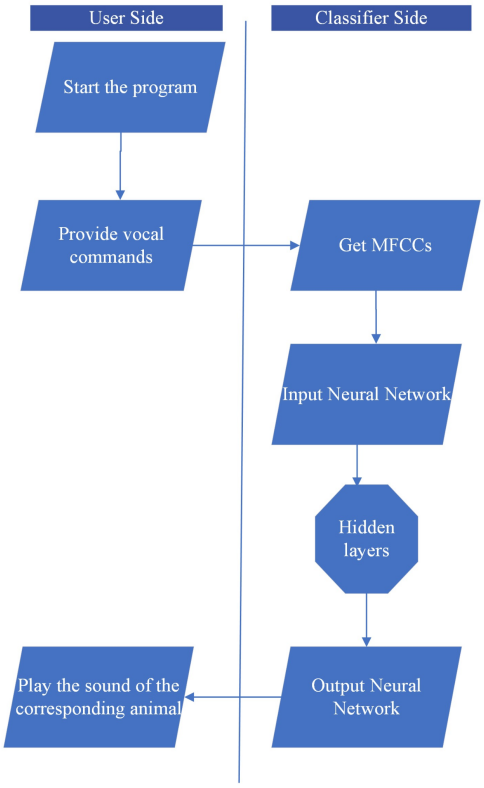


Diagram 2: Activity

System Design

Components of System

Hardware

- Microphone: The microphone functions as a sensor that gathers auditory input.
- Speaker: The speaker functions as an actuator and outputs audio based on the result of the classification of the input.
- Computer: The computer processes the data gathered by the microphone, runs the classification algorithm and in turn tells the speaker what to output.

Software

- Matlab including VOICEBOX toolbox: Used to create all our scripts.
- Classification algorithm
- Processing scripts

Classification algorithm

We started by going through our options for the classification algorithm. We immediately ruled out the rule based algorithm since we found that it would be hard to find a feature with enough discriminatory power and a corresponding rule set to create such a classifier. We also felt as though it would be hard to add new classes (and thus new animals) to this form of classifier. We ended up concluding that both the template matching approach as well as the neural network approach would be viable for our purpose. Due to the time restraints and due to our prior experience we decided to use the neural network approach as the classification algorithm. We plan on trying out different amounts of hidden layers to try and increase the accuracy, we will try to find a balance between the accuracy gained and the computational speed lost.

Features

We found that for our purpose we could either use the frequency spectrum extracted from the input, or we could use the Mel-frequency cepstral coefficients generated from the input. Initially we felt as though the frequency spectrum would have been sufficient. Following the literature study we found that the frequency spectrum would not fulfill the requirement of accounting for variances of the input and therefore we decided on using the Mel-frequency cepstral coefficients. We generate these by giving the sample and the sampling frequency as inputs to a function provided within the VOICEBOX toolbox.

Processing

For the processing we decided on creating scripts that would take a sample, generate its Mel-frequency cepstral coefficients (first 12 of each frame) and sort the data in the correct format

as input for the classifier. Additionally since we only want to respond to an input when a human is speaking we can use a passband filter to filter out the background noise. We found that a passband of 80-400 should encapsulate all possible ranges of signals that are possible to be produced by humans, both adults as well as children. It remains to be seen if the usage of such a filter improves the results and thus we want to further investigate this assumption.

Planning

Schedule

We found that most parts of the project were dependent on earlier parts and thus we went with a waterfall model approach. We set deadlines that should be done during the lab and deadlines that should be done within a timeframe.

Date	Who/What
3/10/2023 (lab)	(All) Plan out the project, get the general idea and the general schedule down.
3/10/2023-6/10/2023 (home)	(Alara + Ana) Do literary study + Write concept of operations. (Yasin) Write code for recording the samples + Write intro plus project goal.
6/10/2023 (lab)	(All) Record samples. (All) Brainstorm for the system requirements.
6/10/2023-10/10/2023 (home)	(Yasin) Scripts for data preprocessing + Write down: system design, project plan with schedule and system requirements.
10/10/2023 (lab)	(Alara + Ana) Start coding the classifier.
10/10/2023-13/10/2023 (home)	(Alara + Ana) Continue coding the classifier and start some testing.
13/10/2023 (lab)	(Alara + Ana) Finish testing and write down the results. (Yasin) Write the evaluation.
13/10/2023-15/10/2023	(All) Make sure everything is in order.

Table 1: Our Planned Schedule

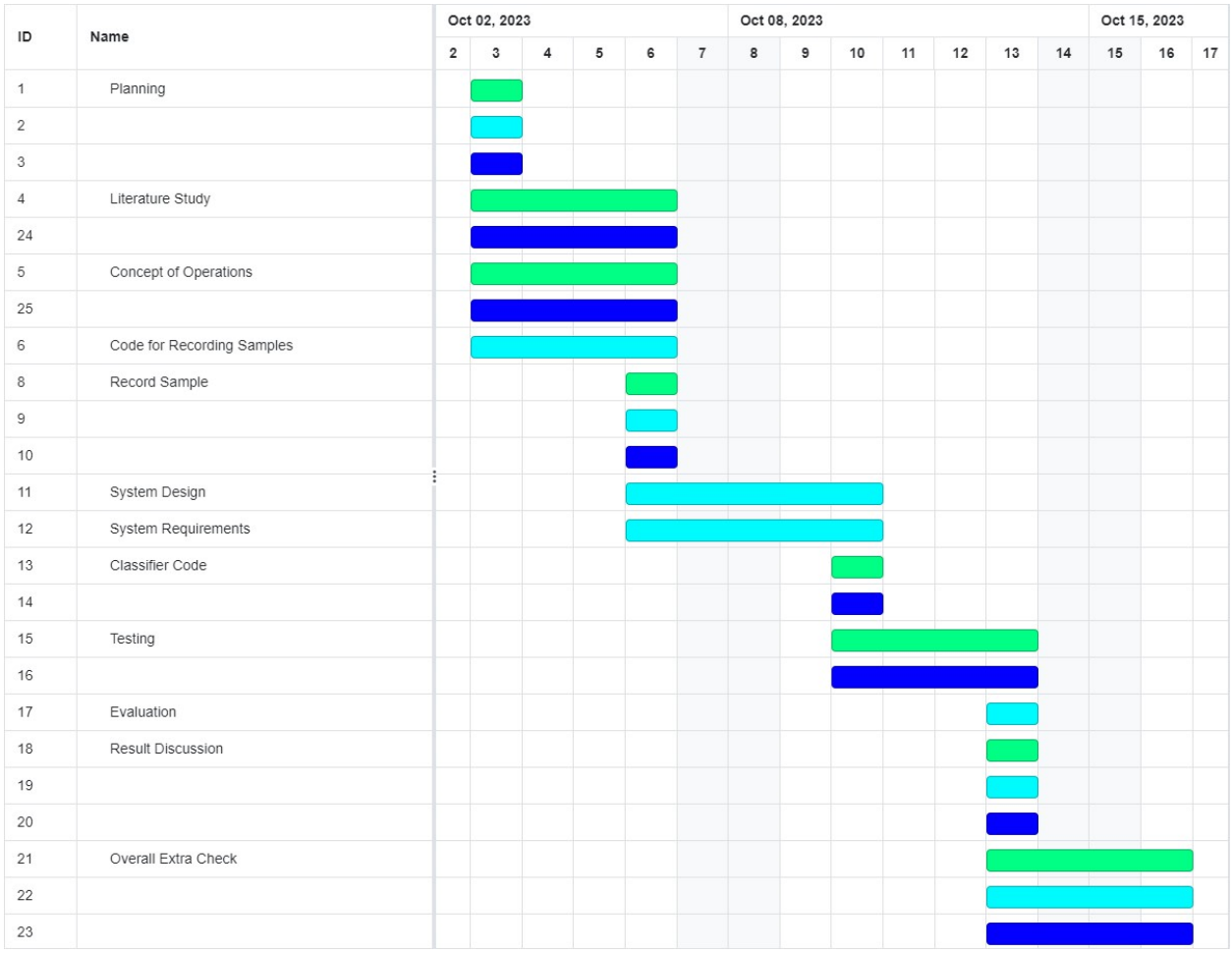


Diagram 3: Gantt Chart of Our Schedule (where each color represents the workload of each member: green for Alara, light blue for Yasin and dark blue for Ana)

Risk analysis

1. Since some parts are dependent on others it might occur that a previously needed component has not been finished and thus the whole project is delayed. In order to prevent this from occurring we prioritized the tasks each person would do. This way we aim to complete tasks that others would need urgently first before moving onto tasks that would not be required urgently for the next part of the project.
2. The workload might be too much at times due to the ambiguity of finishing on the 15th. We can mitigate this by being a bit flexible with the deadlines and setting the final deadline to be one or two days later. We will start each session with a quick recap on what has been completed and what still needs to be done, this way we will not run into an issue where one person is delayed too much without the rest of the group knowing about it.
3. We might lose the samples or the created scripts, to this end we should record the samples into audio files and the scripts into matlab functions. These we will save on a cloud such

that it would be easy to work on multiple devices and such that there would be no way of losing the valuable data.

Testing

Filter or No Filter

We had the idea to use a filter on the input in order to help clear out the background noise. In theory the use of a filter would give higher accuracy, especially in more crowded areas. We tested this hypothesis by training two separate neural networks and comparing the resulting accuracies. Both would be given processed data as inputs but only one of them would be trained on data which had been filtered using a passband of 80-400 HZ . We used 5% validation, 5% testing, and 90% training data to test these out, we also used 25 hidden layers in both of these tests to keep it consistent.

With Filter:

Image 1: Confusion Matrix with filter

As we can see, using the filter has given us 94.4% accuracy on the confusion matrix

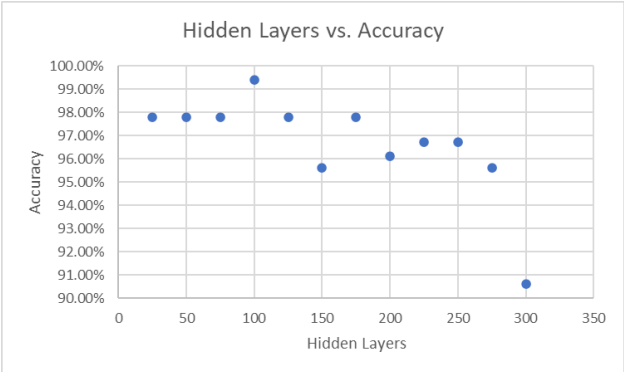
Without filter:

Image 2: Confusion Matrix Without Filter

Looking at the accuracy of using no filter, we see that we get 99.4% accuracy, which means that it is better to use no filter as the usage of a filter ends up decreasing our accuracy.

How Many Hidden Layers?

Now that we concluded not to use a filter we additionally wanted to test out multiple values for hidden layers, we did so in order to maximize the accuracy. To this end we tracked the accuracy of multiple neural networks whereby the only difference would be the amount of hidden layers they contained. While conducting these tests we decided to use a training data set of 70%, a validation set of 15% and a testing data set of 15%. The reason for using bigger data sets for testing and validation is to get more accurate confusion matrices, that have bigger differences in the accuracies. The sizes of these data sets have been kept consistent throughout the testing. For the final neural network we have used a 100% training data set.



Graph 1: Hidden Layers vs. Accuracy

Hidden Layer	Accuracy
25	97.80%
50	97.80%
75	97.80%
100	99.40%
125	97.80%
150	95.60%
175	97.80%
200	96.10%
225	96.70%
250	96.70%
275	95.60%
300	90.60%

Table 2: Hidden Layers vs. Accuracy

The time it takes from input to output is so small in all these cases that we can conclude that there is no additional benefit in accounting for computational speed in our decision of using a certain amount of hidden layers. Thus we concluded that accuracy alone was our main deciding factor and thus we ended up using 100 layers for our final product since, as can be seen from both the graph and the table, the best accuracy we found is achieved at 100 hidden layers.

Unrelated people

Lastly, after incorporating the previous conclusions, we wanted to test out the following requirement: “The system shall have the capability to classify the inputs regardless of the person.”. We did so by asking family and friends to try out our now finished prototype and measured the results by keeping track of the correct outputs vs the incorrect outputs. We found that with the word cow there were some misclassifications. Sometimes the classifier would classify the word cow as cat or dog. We concluded that this may be due to the overlap of the “c” sound with the word cat, and the overlap of the “o” sound with the word dog. Otherwise, with other words we have not noticed any problems and the classifier did not run into any issues. Thus we conclude that the requirement has mostly been fulfilled.

Evaluation

We found that the contributions by each individual was about equal in the end, this was due to the team work whereby in the case that we ran into issues we all helped solve the problems together. This ended up making the project manageable and caused us to finish well before the final deadline. We did end up finishing slightly outside our own expectations and found that this was due to there not being any time planned to deal with issues that came up. We deviated from the Gant chart and the general schedule when we started the testing process since this is when we found out that we made mistakes with the processing. This caused some delays in some deadlines we set in the schedule. In the future we would like to plan some additional time to deal with problems that come up, in our case we did account some time for these kinds of scenarios but we found that the time we planned ended up being too little.

We found that the waterfall approach did help us since we all had some form of individual tasks, though we found out that we did not explicitly plan meetings to check up on the progress of these tasks. In the future we would like to do so since it would give more clarity on the exact schedule, that way each member can also have a clearer understanding of what and when something will be done. We ended up planning these meetings outside of the document schedule which caused some confusion. Due to the importance of these meetings we will include them in the initial planning of future projects.

If we were to have used FFT coefficients as inputs to our classifier we most likely would not have fulfilled the requirement of having the capability to classify the inputs regardless of the person. Thus we made the correct decision to switch to MFCC as input, though we did run into some issues with this approach. We correctly identified and implemented the general approach but found that the classifier had an unexpectedly low accuracy, we found that this was due to the frames from which we took the MFCC from. We initially took the MFCC of all the frames and thus were not only giving the MFCC of frames where the actual words were spoken, but also frames whereby we only had background noise. This caused there to be timing issues and an incorrect training of the neural network. We fixed this issue by looping through the audio data and cutting the first part of the audio sample where no speaking was done, this part only contained background noises and thus these frames were of no importance. Afterwards we gave the cut sample to the MFCC function and chose to only take the first 12 MFCC of the first 15 frames. This is since we identified that 15 frames should be about the time that it takes to say most of the words. Though we also felt as though this number of frames could have been increased to 20, this might have fixed the issue we ran into with the classification of the word “cow”.

Lastly it would have been a better choice to include a more systematic way of measuring the accuracy of the real world performance when we did the testing in the “Unrelated People” paragraph. We should have created our own confusion matrix and should have kept track of the

results after which we could have accurately determined the real world performance of the system. This would have let us know if we were actually facing problems with the classification of the word “cow” and if so what the accuracy was. It would have given us important insights on the classifier as a whole and might have unearthed additional issues we had no clue of. Unfortunately due to time constraints we found that we did not end up having sufficient time to do these measurements systematically. In the future we would like to prioritize the testing since we found that the real world performance should have been more important to us than other tests we performed and we found out that we should always systematically measure performance in order to accurately measure performance.

Bibliography

[1] Juan Colonna, Tanel Peet, Carlos Abreu Ferreira, Alípio M. Jorge, Elsa Ferreira Gomes, and João Gama. 2016. Automatic Classification of Anuran Sounds Using Convolutional Neural Networks. In Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering (C3S2E '16). Association for Computing Machinery, New York, NY, USA, 73–78. https://doi.org/10.1145/2948992.2949016

[2] M. S. Arun Sankar, M. Aiswariya, Dominic Anna Rose, Bhat Anushree, D. Bhagya Shree, P. Mohan Lakshmipriya, and P. S. Sathidevi. 2018. Speech Sound Classification and Estimation of Optimal Order of LPC Using Neural Network. In Proceedings of the 2nd International Conference on Vision, Image and Signal Processing (ICVISP 2018). Association for Computing Machinery, New York, NY, USA, Article 35, 1–5. https://doi.org/10.1145/3271553.3271611