Requirements Document

Client: Zeehondencentrum Pieterburen

Andrei Mihalachi *s3491862*
Annika Moeller *s3585832*
Brian de Jager *s4104579*
Mike Lucas *s3519120*
Roan Rosema *s4109449*

Lecturer: Dr. Andrea Capiluppi
Teaching Assistant: C.M. Rosiu
March 30, 2021

Contents

1 Introduction

The vets who work with the rescued seals at the Zeehondencentrum Pieterburen use several methods to assess the health of the seals once they have been brought to the sanctuary. The goal of this application is to create a tool that will make use of artificial intelligence technology to aid them with this process. Application XYZ is a Windows application that uses a convolutional neural network (CNN) to analyze the health level of a seal based on the sounds it makes. The user can upload a sound file of a seal making a noise, which is analysed by the CNN. The CNN then outputs a score from 1-3 (double check this) indicating the seal's health level (elaborate once CNN details are known).

This information is output in a table format, with the rows indicating the name of each sound file and its corresponding health score. Each row will also have a space for other relevant annotations to be entered by the vets (e.g temperature, bodyweight, general behaviour, etc). The user will also have the option to add more sound files once the table has already been generated.

The data generated by the CNN should be portable, so the application will allow the user to download the generated table as a .csv file. Furthermore, the application will also allow users to upload a previously generated .csv file, view it in the application window, and continue editing the table.

2 Actors

- **Target user**: This application is being developed for the vets who work at the seal sanctuary of Pieterburen
- Possibly other seal sanctuaries

3 Use Cases

In the upcoming sections we elaborate on the most important use cases of our program.

3.1 UC1: Upload .wav audio file

Corresponding user story

As a user, I want to upload audio files to let the system analyze it.

Rationale/Context

The veterinarians at Zeehondencentrum Pieterburen want to analyze the survival prospects of their seals. This can get done by results returned by the system. The user being able to upload audio files is therefore one of the main use cases we should deal with.

Frequency of Occurrence

A few times a month (could vary depending on when analyzation is done)

Primary Actor

The veterinarians at Pieterburen are the primary actors for uploading audio files.

Preconditions

The user has access to the system.

Postconditions

The uploaded audio file is added to the back end with the result generated from the CNN model.

Main Success Scenario:

1. The User asks the system to upload an audio file.

2. The System gives the User a file chooser.

3. The User chooses a .wav audio file and sends it to the System.
4. The System parses the audio file.
5. The System uses the CNN model to generate the likelihood of the seal surviving.
6. The System sends the gathered information to the back end of the System.

3.2 UC2: Export .csv file

Corresponding user story

As a user, I want to export/download a .csv representation of the processed audio files.

Rationale/Context

When audio files have been processed and the data of these files have been stored in the system, the user might want to download this data and work on another set of data or to use the data set on a different computer. The user should therefore be able to export this into a .csv file. The user can continue working on that data set when opening the downloaded .csv file into the system.

Frequency of Occurrence

Possibly a few times a month.

Primary Actor

The veterinarians at Pieterburen are the primary actors.

Preconditions

The user has access to the system and at least one audio file has been uploaded to the system.

Postconditions

The data of the audio files have been exported to the computer as a .csv file.

Main Success Scenario:

1. The User asks the System to export the data
2. The System generates the .csv file
3. The System returns the .csv file to the User

3.3 UC3: Upload .csv file

Corresponding user story

As a user, I want to upload a .csv representation of the processed audio files.

Rationale/Context

The user might want to upload an already existing .csv file to the program. For example, when a user wants to add information about additional analyzed .wav files to an already existing .csv file, a user might want to upload it. Or if a user has forgotten to add one missing sound file, and finds out after already started working on a new session. Also, a user might want to alter data inside the table of an older session. Therefore, uploading a .csv file is a use case needed to be incorporated.

Frequency of Occurrence

Possibly a few times a month.

Primary Actor

The veterinarians at Pieterburen are the primary actors.

Preconditions

The user has access to the system and a .csv file formatted conform the required format of the application.

Postconditions

The application opens a session with data from a already existing .csv file

Main Success Scenario:

1. The User asks the System to upload the csv file
2. The System checks if data in .csv file is structured conform the structure of the program
3. The System opens a session with data from existing .csv file

3.4 UC3: Add a column to table/data

Corresponding user story

As a user, I want to add columns to keep track of more information

Rationale/Context

The user might want to write down and store additional pieces of information which is not included by the application. For this reason, the user wants to add one or more columns with headers that satisfy their need of reserving specific information in an organized fashion.

Frequency of Occurrence

Possibly a few times a month.

Primary Actor

The veterinarians at Pieterburen are the primary actors.

Preconditions

The user has access to the system and a session is currently running.

Postconditions

The application is running a session with one or more columns in the data manually added by the user.

Main Success Scenario:

1. The User asks the System to add a column to the data
2. The System asks the user for the name of this column
3. The System adds a new column to the table and updates GUI
4. The System updates the data in the back end to keep track of this additional data

4 Requirements

4.1 Critical Functional Requirements

1. Upload one or more audio files at a time

   - The user should be able to upload a sound file (in .wav format) to the application. The 'upload' button should lead to a pop-up menu that allows the user to select only .wav files from their computer.

2. Pre-process the uploaded audio file(s)

   - The audio files must be pre-processed such that they are in a format that the CNN can read. They should be resampled using a sampling rate of 8000Hz, and the Mel-Frequency-Cepstral-Coefficients (MFCCs) should be retrieved.

3. Apply the CNN model to the processed audio file(s)

   - The pre-processed sound file must be passed to the neural network so that it can generate the seals' health score.

4. Display the output of the CNN

   - The application must show the output of the CNN (i.e the health score) in a table format.

4.2 Important Functional Requirements

- Add additional information of the seal to the notes section of the table.
- Export the data gathered within the application's table to a .csv file.
- Upload multiple audio files at the same time.
- Upload another .wav file to table once it has already been generated.
- Upload and edit a previously generated .csv file.
- Add columns to the table.

- Delete row from the table.
- Export spectrogram.
- Export the CNN model's input.
- Display the type of uploaded sound file.

4.3 Useful Functional Requirements

- Download newly generated audio files
- Download the generated output
- Make the program more modular (accept more types of weights)
- Generate statistics based on the result of the CNN model
- Put left and right lung of a seal in the same row
- Delete columns from the table.

4.4 Non-Functional Requirements

- Run on a Windows computer
  - The application must work on Windows.
- User-friendly
  - The user interface should be clear and easy to use
- Professional looking interface

4.5 Won't do

5 Change Log

Who	**When**	**Which section**	**What**
Annika	10-02-21	The document	Created the document
Brian	11-02-21	Requirements	First draft of requirements
Annika	12-02-21	Requirements	Updated requirements
Annika	12-02-21	Introduction	First draft
Brian & Roan	16-03-21	Use Cases	Added UC1 and UC2
Brian	29-03-21	Requirements	Added new requirements
Roan	29-03-21	The document	Small restructure + target users updated
Mike	30-03-21	Use Cases	Added UC3 and UC4