Minor Applied Data Science

Research Paper

Detecting Eating and Drinking sounds

Maarten de Jonge, 17113571
Florian Rumpold, 21084424
Lucas Pawlak, 21120854

The Hague University of Applied Sciences

January 2022

Detecting Eating and Drinking sounds

1. Introduction

In the modern world, ever more people are affected by dementia (Alzheimers Association, 2021). Meanwhile, the capacity for care is not increasing at the same rate (Hinkema et al., 2019). Because of this, scientists are looking into alternative ways of taking care of people, mostly seniors, who suffer from dementia. One such project is the Smart Teddy Bear Project (THUAS, 2021).
The aim of the Smart Teddy Bear Project is to create a teddy bear containing sensors, which monitor the surroundings of the teddy and the seniors with dementia. By having the teddy monitor the quality of life on a day-to-day basis, the load on the caretakers is reduced.
The teddy works by using several different sensors that are constantly collecting data. It then uses this data, along with machine learning and deep learning algorithms, to generate a numerical value for several factors that contribute to an overall quality of life score.
Our project group was tasked with identifying eating and drinking sounds in a stream of audio. To accomplish this, several deep learning algorithms are employed.
This paper will answer the question “Which deep learning model is optimal to detect eating and drinking sounds from audio?” An optimal model, in this case, is a model with high accuracy and precision. To answer the question, this paper will first discuss the used dataset and the steps that were undertaken to augment and balance it. The paper will then discuss several different algorithms that were used to try to detect eating and drinking sounds. These include Linear models, Convolutional Neural Networks and Transfer learning. It will compare the accuracy- and precision scores of all of these models against each other and finally it will explain why the final choice of model went to a Convolutional Neural Network and recommend some next steps to take in the continuation of this research.

2. Dataset

The Dataset for this project was obtained through Google’s AudioSet. It contains 1,789,621, 10 second long audio samples from YouTube videos, split into 632 labels (Gemmeke et al., 2017). Among them are 829 Chewing and mastication sounds (*AudioSet Chewing, Mastication 1*, n.d.), which were used as positive samples. The negative samples were chosen to be mainly domestic sounds: vacuum cleaner, typing, speech, door, toilet flush, clapping, silence, knock, wind, background noise, sink. In total 1544 negative samples were used.

The Drinking sounds were chosen from a Dataset of a Kaggle Competition. (*Eating Sound Collection*, 2020). It contains 291 Drinking sounds which were then overlaid on random negative samples from Google’s AudioSet, in order to get a uniform length of 10 seconds for the drinking sounds.

A Github Repository called ‘audioset-Processing’ (*Aoifemcdonagh/audioset-Processing: Toolkit for Downloading and Processing Google's AudioSet Dataset.*, n.d.) provided the functionalities to download the chosen audio files. After the download it was discovered that

only 224 chewing sounds have successfully been downloaded. The reason for this is that most of the videos do not exist on YouTube anymore.

Figure 1 shows the distribution of the labels in a histogram. This Dataset is unbalanced, but we applied Data Augmentation to balance it, as explained in the Data Augmentation chapter.

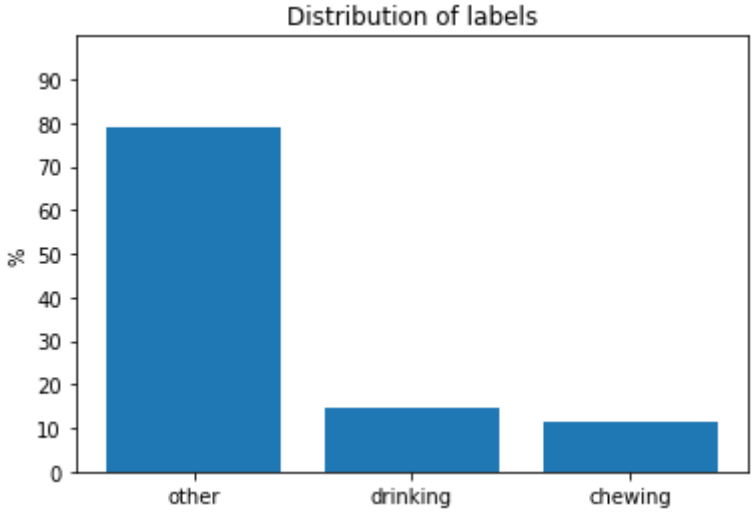

Figure 1: Histogram of labels in unaugmented dataset

Of a random sample of size 10 of the Chewing sounds, 90% of the samples were accurate (*AudioSet Chewing, Mastication*, n.d.). After taking a random sample of size 10 of the drinking sounds, it was found that 100% of the labels were accurate. A random sample of 20% was used as validation data.

2.1. Data augmentation

To balance the dataset, as well as increase the total amount of samples, the positive samples were altered slightly in several ways. All alterations were applied to all positive samples, but only to a number of negative samples. These alterations include:

- shifting the sound forward in time (and padding the beginning with zeros),
- shifting the sound backwards in time (again padding with zeros),
- pitching the sound up,
- pitching the sound down,
- stretching/slowing the sound,
- squeezing/speeding up the sound.

After each of these alterations, the sound files are cut to be exactly 10 seconds long or, if they are shorter than that, as with the squeezed samples, they are padded with zeros. This way the model only encounters sounds of equal length, ensuring it doesn't take file length into account during training.
The negative samples are also augmented, to make the models train according to sounds and not according to the alterations made. For the same reason the unaugmented samples are still included in the final dataset.
Figure 2 shows the distribution of the labels in a histogram after the Data Augmentation.
In total there are 7832 samples, 1792 chewing sounds and 2328 drinking sounds.

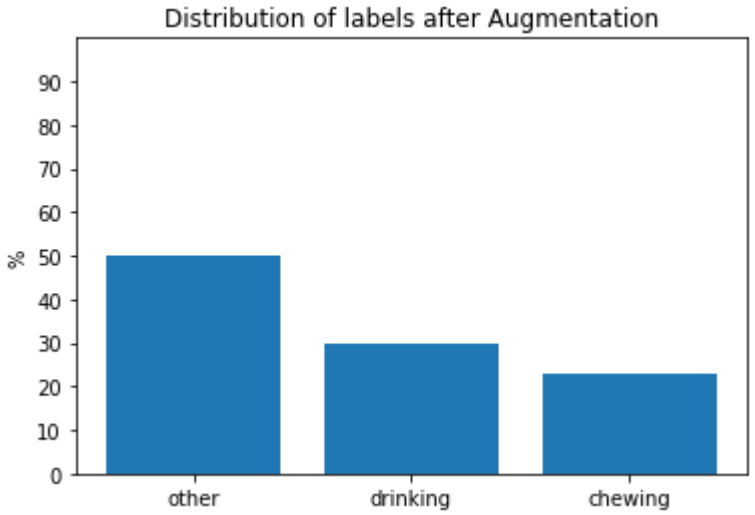

Figure 2: Histogram of labels in augmented dataset

2.2. Data preprocessing

To normalise our current dataset, multiple data preprocessing techniques were used. The Torchaudio library (Yan et al., 2021) was used for the preprocessing.
As the Torchaudio.load function uses the native sampling rate of the file, we had to resample all the sound files to have the same sampling rate on the whole dataset.
We made every sound file exactly 10 seconds long, because most of the files on the dataset were around 10 seconds already. The final model will be receiving a continuous audio stream, which can be cut into 10 second fragments. For predictions 10 second intervals will also be used.

We resized every sound file to 10 seconds either by adding silence or cutting. The longest sound file was 10.001 seconds, so there was practically no loss of information.
We turned the sound files into a Mel spectrogram with a sampling rate of 16 000 Hz, n_fft (the number of fast Fourier transforms) set at 400, and n_mels (number of frequency bins) set at 64.

3. Architectures

We explored which type of model performed best, given that they have the same hyperparameters. We chose 3 architectures for evaluation: Linear models, Convolutional Neural Network's (CNN) and Transfer Learning with models from ResNet. Linear models were first used as a benchmark. Because ResNet models and CNN's are capable of classifying audio (Hershey et al., 2017), we implemented and trained several different ResNet models and CNN's. The models were implemented using the PyTorch (Paszke et al., 2019) library. The output layer of the models produces probabilities of the predicted classes. The class with the highest probability is selected as the prediction.

3.1. Setup Hyperparameters

To create an environment where all models have the same or comparable prerequisites, hyperparameters had to be constant. We decided to use a learning rate of 0.01 and a batch size of 64. Each model was trained for 100 epochs. Since we were dealing with a multiclass classification problem, CrossEntropyLoss was used as the loss function. For training, the AdamW optimizer was used. To balance the training Data Loader we added a *WeightedRandomSampler* with normed label weights.

For all of our self-implemented linear models and convolutional neural networks the same number of hidden nodes per linear layer was chosen to be 100.

3.2. Linear Model

In total 3 different models with solely linear layers were trained. The number of hidden layers ranged from 0 to 2.

3.3. Convolutional Neural Network

We implemented and trained 2 groups of CNN's. Both groups consisted of 6 models. The models in each group differed in the number of fully connected hidden linear layers, ranging from 0 to 5. The first group had 1 convolutional layer with 8 out channels. The second group had 2 convolutional layers, where the first convolutional step had 8 out channels and the second convolutional step had 16 out channels.

3.4. Transfer Learning Using torchvision's ResNet Models

For transfer learning the pretrained ResNet18, ResNet34, ResNet50, Wide ResNet50-2 and the ResNet152 of the torchvision (Marcel & Rodriguez, 2010) package were trained. We had to modify the first convolutional step in all models to fit our data. The output layer was also modified to have 3 nodes to fit the number of labels in our Dataset.

4. Model evaluation

While training, the epochs were sorted based on overall accuracy. The epoch with the highest accuracy was deemed the best version of the model. The other metrics (precision and recall) were calculated for every epoch as well. All of these metrics were taken into consideration in deciding the best model. In the case of similar accuracy, precision was used as the deciding metric. To diagnose the model, the validation loss and accuracy scores were plotted.

5. Results

Figure 3 shows the metrics of the best performing model of each architecture. The results are described in detail in the next chapters.

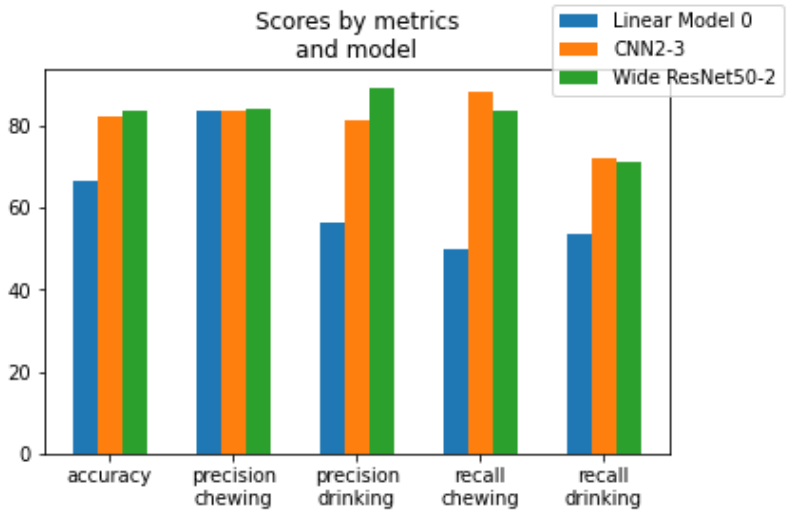

Figure 3: Scores in different metrics for the best linear, CNN and ResNet models

5.1 Linear Model

The best linear model had no hidden layers. On the best epoch it achieved an accuracy of 66.4%, a precision for chewing of 83.5%, a recall on chewing of 50.1%, 56.2% precision on drinking and 53.8% recall on drinking on the validation data. The accuracy scores for each epoch are shown in figure 4.

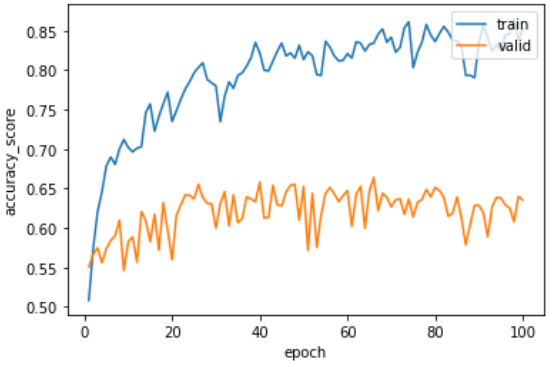

Figure 4: Plots of loss and accuracy against epochs of Linear model with no hidden layers

5.2. Convolutional Neural Network

The highest accuracy a self implemented CNN achieved was 82.2% on the validation data. Precision and Recall on chewing sounds were 83.5% and 88.2% respectively. For drinking it achieved a precision of 81.4% and recall of 72.1%. This CNN consisted of 2 Convolutional Steps and had 3 hidden linear layers. The accuracy scores for each epoch care shown in figure 5.

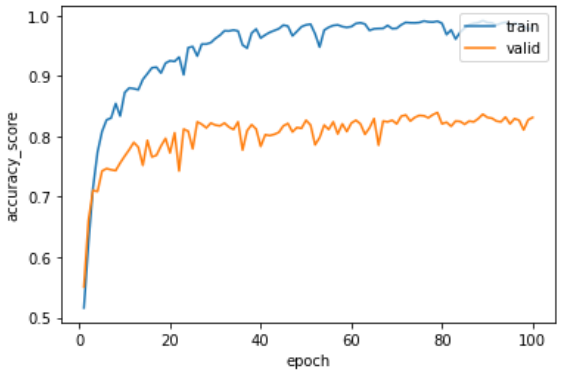

Figure 5: Plot of accuracy against epochs of the CNN with 3 convolutional steps and 3 hidden layers

5.3. ResNet Models

Of all the ResNet models that were trained, the Wide ResNet 50-2 achieved an accuracy of 83.6% on the validation data. Precision and recall for chewing was 83.8% and 83.6%. For drinking precision and recall were 89.1% and 71.0%.
The loss and accuracy on the validation data was oscillating heavily, as shown in the figure 6.

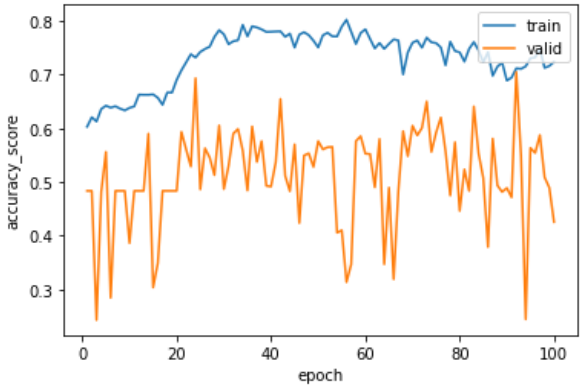

Figure 6: Plot of accuracy against epochs of the Wide Resnet 50-2 model

5.4. Deciding on an Architecture to finetune

The solely linear models performed worst in every metric in every experiment. For this reason, we decided not to include these models in any further experiments.
In multiclass classification the CNN and the Wide ResNet 50-2 had similar results, with the wide ResNet 50-2 having overall slightly higher accuracy and precision, however the ResNet models fluctuated both in validation loss and accuracy over epochs on the validation data. The reason for this Fluctuation is unknown. The CNN had a much more stable loss and accuracy. Therefore we decided to finetune the CNN to maximise overall accuracy and precision on chewing and drinking.

6. Tuning Hyperparameters

We decided to tune the CNN with 2 convolutional steps and 3 hidden layers. The following hyperparameters were tuned: the number of nodes per hidden layer, learning rate, batch size and weight decay.

The best result was achieved with a CNN that was wider in the middle and narrower in the last layers. Also a linear degrading learning rate seemed to smoothen the learning curve. We noticed no real difference when a batch size of 128 or 256 was chosen. To counter overfitting we used a weight decay of 0.5. In figure 7 the confusion matrix of the final result is shown.

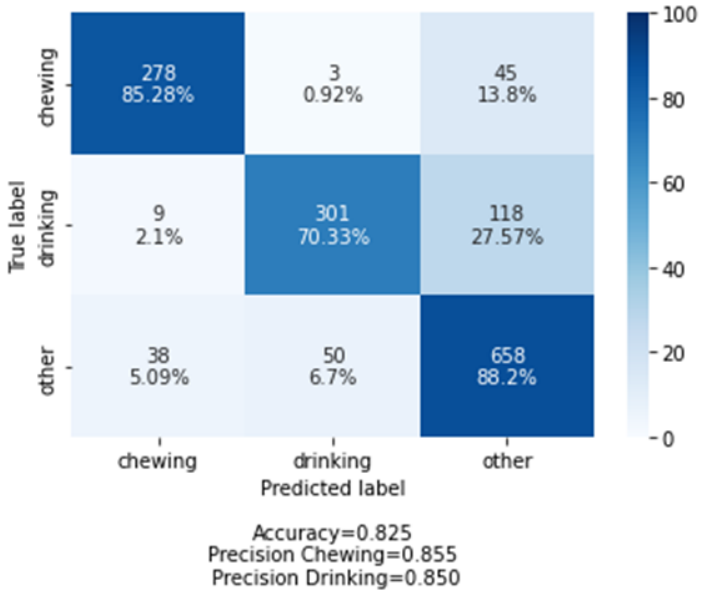

Figure 7: Confusion matrix of the best performing model

7. Conclusion

In trying to find the best deep learning model to detect eating and drinking sounds, several models were used. These can be divided into three categories: purely linear neural networks, convolutional neural networks and transfer learning using ResNet models.
The linear models were the worst performers in all metrics. On its best epoch it achieved an accuracy of only 66.4%.
The self-made CNN's had a good curve in the accuracy and loss graphs and ended on an overall accuracy of 82.5%, a precision of 85.5% on chewing and 85.5% on drinking and a recall of 85.28% on chewing and 70.33% on drinking. As we prioritise precision over recall, these results are very promising.
The ResNet models had a few epochs that exceeded the performance of the self-made CNN's. The best of these achieved an accuracy score of 83.6%. This is 1.4 percentage points better than the best self-made CNN. However, as can be seen in figure 6, the accuracy of the ResNet models varied heavily between epochs. The best performing model had an epoch with an accuracy of 83.6% followed immediately by an epoch with an accuracy

of only 24.4%. Because of this high variance, the transfer learning models are not our recommendation, despite some good performances.
Considering this, our best model for detecting chewing and drinking sounds is a Convolutional Neural Network with 2 Convolutional layers and 3 linear layers.

8. Recommendations

For future research it might be helpful to construct a dataset where the audio was collected directly from the teddy in order to create a dataset that better represents the situations a teddy will be in.
Also including sounds that often occur during chewing and drinking e.g. cutlery sounds might lead to more promising results, since these sounds also occur during a meal.

9. Acknowledgments

We would like to thank Hani Al-Ers, Jeroen Vuurens, Tony Andrioli and Ruud Vermeij for their valuable guidance and feedback during this project.

References

Alzheimers Association (Ed.). (2021). Global dementia cases forecasted to triple by 2050. *Alzheimers Association International Conference 2021*, 4. https://www.alz.org/aaic/downloads2021/Global_Prevalence_plus_younger_onset_and_US_mortality.pdf

aoifemcdonagh/audioset-processing: Toolkit for downloading and processing Google's AudioSet dataset. (n.d.). GitHub. Retrieved December 10, 2021, from https://github.com/aoifemcdonagh/audioset-processing

Audio manipulation with torchaudio — PyTorch Tutorials 1.10.0+cu102 documentation. (n.d.). PyTorch. https://pytorch.org/tutorials/beginner/audio_preprocessing_tutorial.html#loading-audio-data-into-tensor

AudioSet chewing, mastication. (n.d.). AudioSet. Retrieved December 10, 2021, from https://research.google.com/audioset//dataset/chewing_mastication.html

AudioSet chewing, mastication 1. (n.d.). AudioSet. Retrieved December 10, 2021, from https://research.google.com/audioset/ontology/chewing_mastication_1.html

Eating Sound Collection. (2020, June 20). Kaggle. Retrieved December 10, 2021, from https://www.kaggle.com/mashijie/eating-sound-collection

Gemmeke, J. F., Ellis, D. P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., & Ritter, M. (2017). Audio Set: An ontology and human-labeled dataset for audio events. *IEEE ICASSP*. Retrieved October 4, 2021, from https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45857.pdf

Hershey, S., Chaudhuri, S., Ellis, D. P.W., Gemmeke, J. F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R., & Wilson, K. (2017, 1 10). CNN Architectures for Large-Scale Audio Classification. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Hinkema, M., Heumen, van, S., & Wissekerke, van, N. E. (2019). Prognose capaciteitsontwikkeling verpleeghuiszorg. 55.

Marcel, S., & Rodriguez, Y. (2010, October). Torchvision the machine-vision package of torch. *Proceedings of the 18th International Conference on Multimedea 2010, Firenze, Italy, October 25-29, 2010*.

Paszke, A., Gross, S., Massa, F., & Lerer, A. (2019, 12). PyTorch: An Imperative Style, High-Performance Deep Learning Library.

THUAS. (2021). *Smart Teddy*. Data Science Research Group THUAS. Retrieved september 3, 2021, from https://bigdata-thuas.eu/projects/smart-teddy/

Yan, Y.-Y., Hira, M., Ni, Z., Chourdia, N., Astafurov, A., & et al. (2021, Octoboer 28). TORCHAUDIO: BUILDING BLOCKS FOR AUDIO AND SPEECH PROCESSING. 5. https://arxiv.org/pdf/2110.15018.pdf