Sound Recognition with a Humanoid Robot for a Quiz Game in an Educational Environment

R. Tutul[1], A. Jakob[1], I. Buchem[1], N. Pinkwart[2]

[1] *Berliner Hochschule für Technik, 13353 Berlin, E-Mail: rezaul.tutul@bht-berlin.de andre.jakob@bht-berlin.de, buchem@bht-berlin.de*

[2] *Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), 10117 Berlin, E-Mail: pinkwart@dfki.de*

Introduction

The use of humanoid robots is rapidly expanding from children's education to higher education with the development of AI technologies and computing hardware that motivate learners and increase their engagement with learning material [1]. Considerable amount of work has already been done to equip robots with visual perception [3]. As a result, robots can model their environment, navigate safely, and recognize people and their everyday actions, gestures, and facial expressions [2]. However, visual perception using image processing has some limitations, for instance, it may not work in low or bright light conditions, and interaction is inherently limited to objects and people within the field of view. Compared to visual perception, audio perception is complementary. Sounds emanating from objects, people, or human-object interactions contain a significant amount of information about the current environment and events that augment the robot's visual perceptual capabilities. For example, by recognizing sound events, a robot can assess their relevance and take corresponding decisions, even if they are not in the field of view. Similarly, developing sound recognition capabilities in addition to the visual and verbal capabilities of humanoid educational robots improves their activity and overall performance in the educational environment.

In this study a quiz game was implemented, in which the robot Pepper asks questions, the participant, who wants to answer, presses a buzzer button, the robot decides, which buzzer was pressed (or which buzzer was pressed first in case of both buzzers activated) and awaits the answer. The battery driven buzzers radiate two different and clearly distinguishable artificial sounds. By the manufacturer of the buzzers these two sounds are called “charge” (blue buzzer) and “laser” (green buzzer). Figure 1 shows the two buzzers. The buzzers are not connected in any way to the robot.

Robot Pepper

Pepper is a humanoid robot [4] introduced by Aldebaran Robotics in 2014. It is shown in Figure 2. It has four microphones in its head, a three-dimensional depth camera in its eyes, and a 10.1-inch tablet in its chest with Android operating system to facilitate human-robot interaction. In our experiment, Pepper version 1.9 is used in the implementation of the buzzer quiz game. This version only supports Android (version 6.0) applications to control the robot movement, poses and gestures. There is no additional camera or microphone for the Pepper tablet, as it is connected to the cameras and microphones in the robot’s head.

Figure 1: Buzzer Buttons (“blue: charge”, “green: laser”)

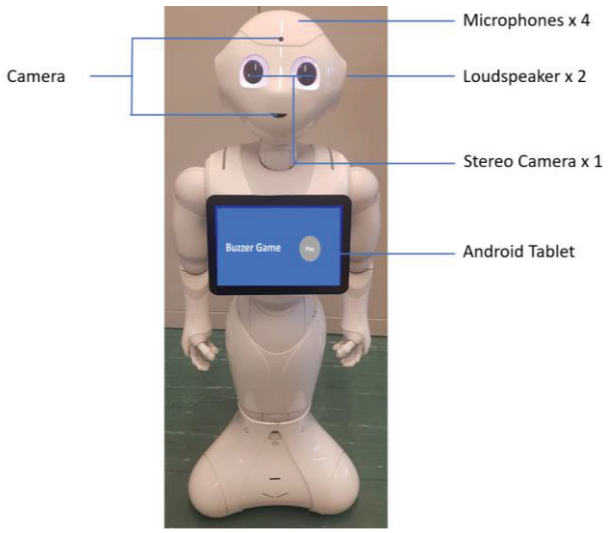

Figure 2: Pepper humanoid robot [5]

Data Set Preparation

For the preparation of the data set the sounds of both buzzers were recorded using the Pepper. The distance between the robot and the buzzers was in the range of up to 3 meters, and all sounds were produced in front of the robot from different positions at a height of about 0.5 to 1.5 meters above ground. The sounds were recorded as single-channel signals with 16 kHz sample rate and 16 bits resolution. In order to produce and record many sounds, each buzzer was pressed repeatedly for approximately 10 minutes. All recordings were made in the same room having a ground area of approximately 20 m².

Audacity was used to manually split and label all the sounds produced, as indicated in Figure 3. With this procedure the data set classes “laser” and “charge” were prepared. One example for each of these two classes are shown in the top two diagrams of Figure 4. These classes shall be recognized in the quiz game, when only one buzzer will be pressed.

From the “charge” and “laser” classes various examples of “charge” and “laser” signals were added together with positive and negative random delays between them in the range of 0.5 up to 1.5 seconds. This yielded two new classes called “chargeFirst” and “laserFirst”, with the name depending on whether the delay between both sounds is

positive or negative. Diagrams in third and fourth row of Figure 4 show one example of each class.

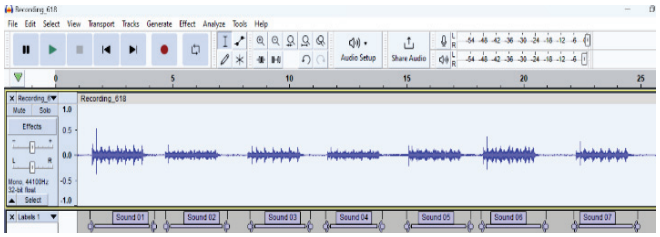

Figure 3: Data split and labeling in Audacity

Additionally, white noise signals were generated for an extra class called “silence” shown in the bottom diagram in Figure 4. A suitable amplitude of the white noise signal was estimated by comparing Pepper’s own noise without buzzer sounds or any other noise. The robot does not stand absolutely still, but is continuously making little movements of its head and arms in order to have some kind of a “live look”. Of course, these movements produce some low noise.

Thus, in total five classes were prepared for the classification algorithm with the following numbers of examples in the data set: “charge” 549, “laser” 447, “chargeFirst” 447, “laserFirst” 447, “silence” 230.

Thus, in total the whole data set consists of 2120 signal examples, each having a length of 3 seconds. The sample rate is 16 kHz, thus, yielding 48000 samples in each signal.

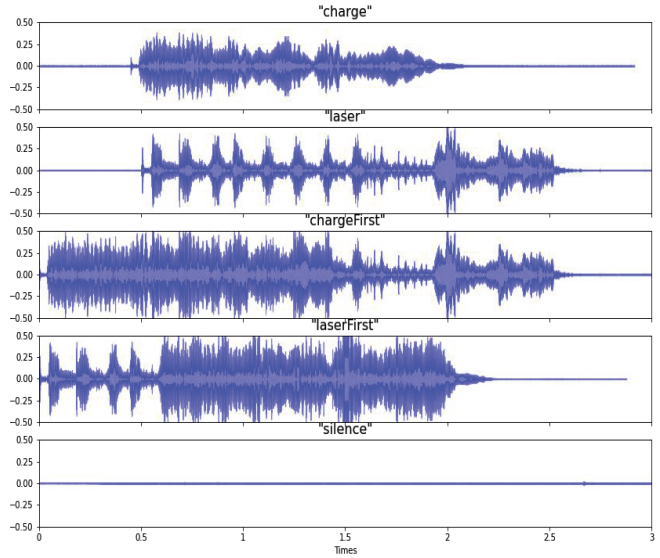

Figure 4: Example signals of all five classes without background noise

The artificially generated sounds of the buzzers are, of course, always played identical. The sounds are mainly only changed by the room impulse response between buzzer and robot depending on the position of the buzzers inside the room during recording. In a practical application however, there will also be disturbances like noise from the quiz participants or from the audience or from outside. To account for this, the signals of all classes were randomly overlaid with five different background sounds including people talking, laughing and more, as shown in Figure 5.

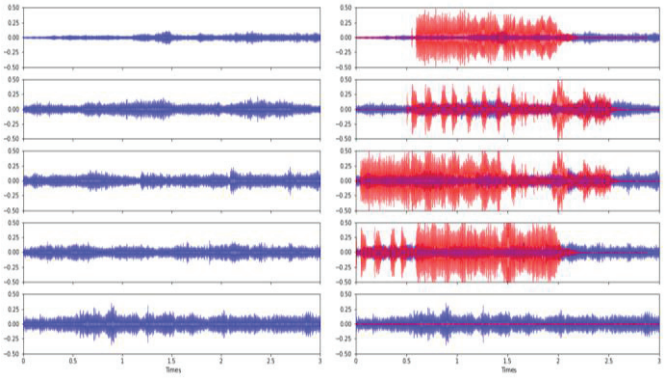

Figure 5: Background noise (blue) and original sounds (red) overlaid with background noise

Sound Recognition

Since years, there is a growing interest in intelligent systems that are able to recognize sounds, e. g. for urban alarm systems, maintenance prediction and also for humanoid robot applications. State of the art has evolved rapidly in recent years with Convolutional Neural Networks (CNNs) widely dominating the field [6, 10]. This is caused by the huge number and success of applications for image recognition. Thus, for sound recognition usually the signal is transformed section by section from the time domain to the frequency domain in order to produce spectrogram-like representations, which are usually transformed into images and presented to the CNN. For the generation of the spectrograms usually either short-time Fourier transform (STFT), mel frequency coefficients (MFC), mel frequency cepstral coefficients (MFCC) or sometimes even wavelet transforms are used.

Here the STFT was used with a frame length of 256 samples, hanning window and a hop-size of 128 samples (“hop-size equals frame length minus overlap”). This yields 374 magnitude spectra each having 129 linearly spaced frequency bins from zero to 8 kHz.

The resulting spectrograms were converted into images. In object or sound recognition research, there is a tendency to collect large amounts of image features to improve performance. However, it is debatable whether the use of more information contributes to higher accuracy, as this involves more computational effort. H. M. Bui et al. [7] revealed classifying with grayscale images led to higher classification accuracy compared to RGB images for the different types of classifiers. We tested both and found similar results with our data set and decided to use grayscale images. Nevertheless, we feel that humans might have a better impression of colored spectrograms. Thus, they are shown here as RGB images in Figure 6, while the network “sees” only grayscale images. In each diagram in Figure 6 time runs from left (0 s) to right (3 s) and frequency runs from bottom (0 Hz) to top (8 kHz) linearly scaled. Magnitude is logarithmically scaled. These 374×129 pixel grayscale images are input data for the CNN.

The CNN used here consists of 3 convolutional layers with 32, 64 and again 64 filters, respectively and ReLU activation functions (rectified linear unit) [8]. Each convolutional layer is followed by a pooling Layer. Max-pooling is used to reduce

the dimension of the previous output and prevent the network from over-fitting with fewer parameters. The third pooling layer is followed by a flatten layer as interface between the convolutional layer part of the CNN and the fully connected layer part. Two fully connected layers ("dense") serve for the classification of the 5 classes to be detected. The CNN model architecture is shown in Figure 7.

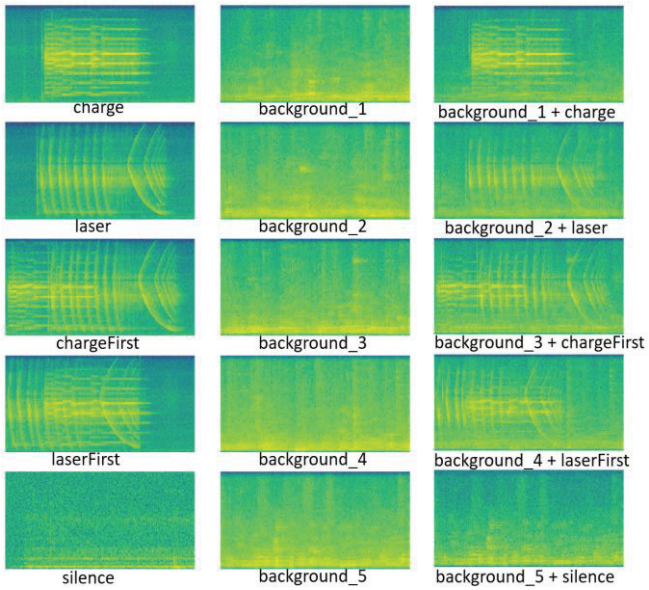

Figure 6: Examples of spectrograms, linear frequency axis, logarithmic magnitude

The data set was automatically divided into 80% training data, 10% validation data and 10% test data. The model was trained over 25 iterations and the Stochastic Gradient Descent (SGD) solver was used with learning rate 0.125.

The validation accuracy was more than 97% and the test accuracy over 95%. In Figure 8 is shown a resulting confusion matrix of the test data, which shows nearly perfect behavior. Such a high accuracy might occur due to the fact that it is pure simulation and the sounds to be detected are always nearly identical except the background noise and changes due to the different source positions in always the same room. It can be expected, that in a practical application with people joining the quiz game or with other disturbances or in another room the accuracy will decrease.

Quiz Game Application

A dynamic quiz game application was developed for the Pepper robot with Android Studio IDE, Kotlin and Python programming language, using Firebase real time database system that enables modification of quiz questions, answers, and time limit of the game. A web-server was developed that uses the CNN model to recognize the recorded sounds by Pepper in a given time limit during the game. The communication between server and Pepper robot was done by the OkHttp communication protocol. The quiz game application architecture is shown in Figure 9 where the Pepper asking and visualizing quiz questions to the game participants from database, offers time slot to press the buzzer buttons and records the buzzer sounds. From the recorded buzzer sounds, Pepper determines the first pressed sound using the developed CNN classification model to take the answer and executes the "Happy" or "Sad" animations [9] shown in Figure 10. The practical test result shows that the model determines the first pressed sound very well, but quite often "charge" is the result instead of "chargeFirst". The same is true for "laser" and "laserFirst". But, for the quiz game to be successful this is not a big problem, because in either case the decision who is allowed to answer is correct.

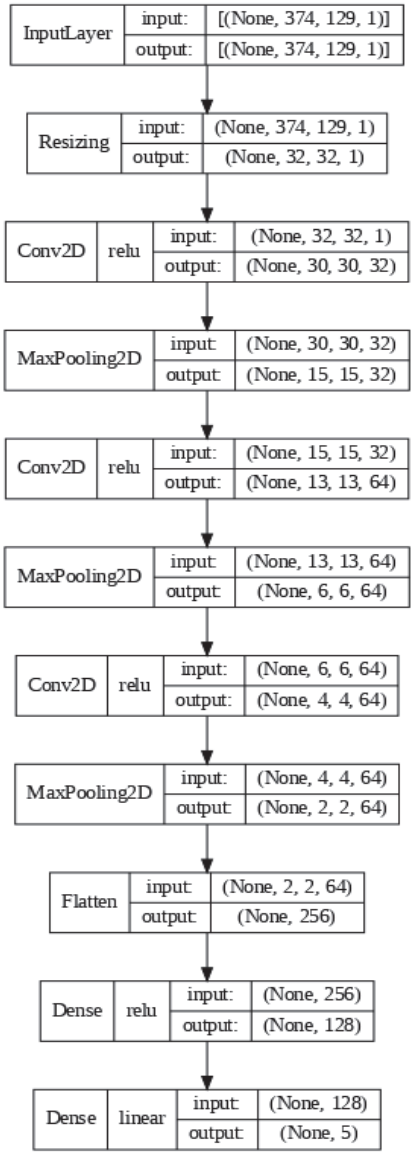

Figure 7: CNN model architecture

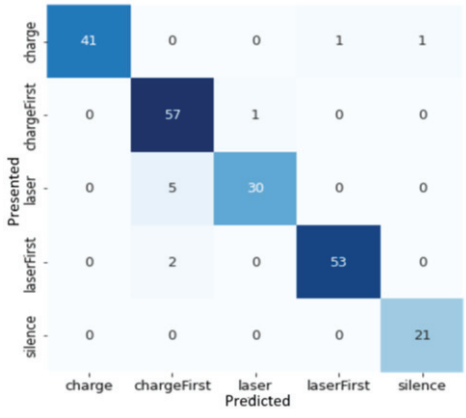

Figure 8: Confusion matrix for test data

Figure 11 shows the user interface of the quiz game application. After pressing the play button, Pepper greets the participants and explains the rules of the game before asking the questions. After reading the questions and answers, Pepper asks the audience to be quiet and the participants to press the keys as soon as the indicator on the user interface changes from white to red. When the indicator turns red, Pepper records the button sounds, and when it turns white again, Pepper displays the recognition result in the bottom part of the user interface. Later, the recognized participant answers and finally Pepper makes the "Happy" or "Sad" animations and shows the score of both participants.

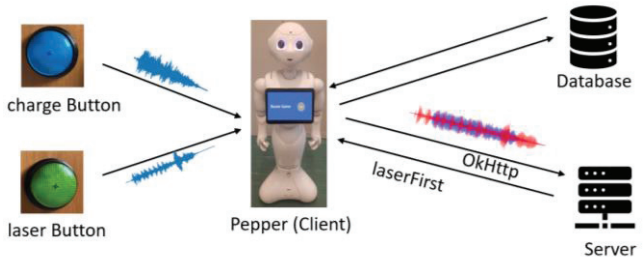

Figure 9: Quiz game application architecture

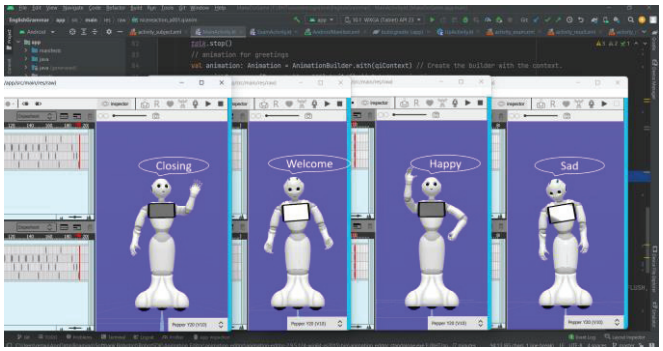

Figure 10: Robot animation of quiz game application

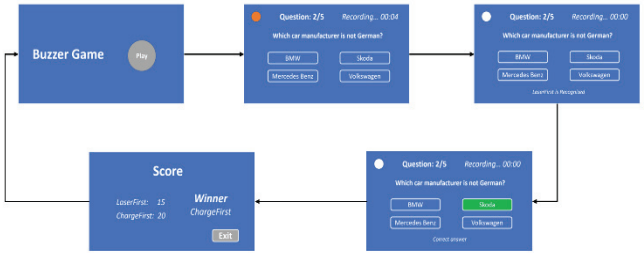

Figure 11: User interface of quiz game application

Conclusion

The technical development has focused on the recognition of sounds, starting with the buzzer sounds used in a quiz game to signal that a participant wants to attempt an answer to a quiz question. The recognition of buzzer sounds was tested and the results show that the CNN can recognize which buzzer sound was pressed first, even when the buzzer sounds are overlaid with a delay of at least 0.5 seconds. Since testing has been done so far without additional sounds, further tests in classroom settings will be necessary and the students learning experience need to be justified by the questionnaires. Further development will focus on recognition of other non-verbal sounds, such as hand clapping, fingers snapping or laughter, in order to detect, affect and enhance motivation, engagement of students towards game based learning with sound recognition using humanoid robots in educational environments.

References

[1] Y. L. Tsai, C. C. Tsai: A meta-analysis of research on digital game-based science learning. Journal of Computer Assisted Learning, 280-294, 2020

[2] N. Chervyakov, P. Lyakhov, D. Kaplun, D. Butusov, N. Nagornov: Analysis of the quantization noise in discrete wavelet transform filters for image processing. Electronics (Switzerland), vol. 7, no. 8, p. 135, 2018

[3] I. A. Hameed, G. Strazdins, H. A. M. Hatlemark, I. S. Jakobsen, J. O. Damdam: Robots that can mix serious with fun. The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018), 723, 595-604

[4] Aldebaran welcome to our support centre page, URL: https://www.aldebaran.com/en/support/topics/welcome-your-new-support-page-0

[5] SoftBank Pepper spec page, URL: https://www.softbank.jp/en/robot/

[6] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, S. Yan: Hcp: A flexible cnn framework for multi-label image classification, IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 9, pp. 1901–1907, 2016

[7] H. M. Bui, M. Lech, E. Cheng, K. Neville, I. S. Burnett: Using grayscale images for object recognition with convolutional-recursive neural network, 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), Ha-Long, pp. 321-325, 2016

[8] V. Nair, G. E. Hinton: Rectified linear units improve restricted boltzmann machines, in Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814, 2010

[9] I. Buchem, M. A. Elroy, R. Tutul: Designing and programming game based learning with humanoid robots a case study of the multimodal Make Or Do English grammar game with the Pepper robot. 15th annual International Conference of Education, Research and Innovation, 1537-1545, 2022

[10] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber: Flexible, high performance convolutional neural networks for image classification, in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, vol. 22, p. 1237, Barcelona, Spain, 2011