Neural Network and Prior Knowledge Ensemble for Whistle Recognition

Diana Kleingarn*(✉) and Dominik Brämer*

Robotics Research Institute, Section Information Technology,
TU Dortmund University, 44227 Dortmund, Germany
diana.kleingarn@tu-dortmund.de

Abstract. Whistle recognition is becoming an increasingly crucial aspect of RoboCup. Therefore neural networks are being utilized in this field more frequently. They are typically more effective than straightforward conventional approaches but still have flaws in fields that require prior knowledge, as conventional approaches do. In this work, we present an approach that can outperform standalone variants of both methods by fusing prior knowledge of traditional methods with a neural network. Additionally, we were able to keep the composite system runtime efficient on the integrated hardware of the NAO Robot.

Keywords: Whistle recognition · Convolutional neural network · Classical audio processing.

1 Introduction

Robots, in general, have a primary purpose, like, in our case, playing soccer. However, on top of their main goal, they have to react to specific events, such as a whistle. Although it is clear that whistle recognition is not the main focus of soccer robots, not reacting to this referee command is usually punished. Whistle recognition was first introduced in the Standard Plattform League (SPL) to indicate the start of a game in 2015. However, in the beginning, the disadvantage in form of a time delay of 15 seconds of the game controller signal compared to the whistle was only implemented for quarterfinals and onwards. Since then, this time delay has been installed for all whistle signals during the games. Furthermore, with the recent in-Game Visual Referee Challenge [6], reliably recognizing the whistle becomes more and more crucial.

One of the biggest challenges RoboCup teams face for audio processing is noise. Just like humans, soccer robots are supposed to react reliably to a whistle, independent of the surrounding, the room impulse response, or the cheering crowd, including children. Loud people, especially children, bring additional challenges to audio processing. This is described in detail in Section 2 in combination with the dataset we use.

* These authors contributed equally to this work.

Since whistle recognition is relevant for all RoboCup teams, a variety of solutions are already presented or published within the teams' code release. The already existing solutions, described in Section 1.1, all have their advantages but, at the same time, do not cover all critical situations. While some solutions are good at detecting soft whistles, they are also likely to detect loud noise as a whistle. Robustness against noise, however, usually causes a lot of false negatives. Moreover, if a whistle is supposed to be detected in a noisy environment, keeping the balance between false positives and false negatives is more challenging. False recognitions are for sure never desirable during a soccer game, especially with the described disadvantage of not hearing a whistle. These include reducing the time robots have to return to their own half of the field after scoring a goal from 45 seconds to 30 seconds or waiting 15 seconds to receive the referee command for kick-off or a penalty kick. Nevertheless, recognizing something, for example the cheering crowd, as a whistle, which was not a whistle, is even worse because a whistle during a robot soccer game means a goal was scored, so the robots would start returning to their starting position.

In order to create a whistle recognizer that works more independently of the surroundings or the used whistle, we combine a neural network with prior knowledge, such as the pattern in the frequency domain created by a whistle or the overall noise over all frequencies. This approach is promising since the advantages and disadvantages of the neural network and the prior knowledge method compensate each other. In addition, the dampening factor ensures that the system is less likely to detect false positives in very loud environments, in which it is also hard for humans to differentiate the sounds. The setup of the proposed solution is described in Section 3. For the evaluation, in Section 4, we test our solution simultaneously with the code release of two other SPL teams and our old whistle recognizer, which uses a neural network. The final section summarizes the results and provides an overview of future projects.

1.1 Related Work

Even though whistle recognition is a pretty specific topic, it is not completely new. For example, the whistle recognizer presented by Bonarini et al. [3] was designed to detect not only a whistle event but also the pattern to identify referee commands. Although pattern recognition is not yet needed in the SPL, other requirements, such as robustness of the detection and against noise or runtime, are similar. They use a frequency mask followed by a neural network approach. In addition, they provide a software tool to help the user to tune the system.

In 2014 Poore et al. [13] introduced two real-time sound recognition approaches designed for the NAO Robot in various noisy environments. The first approach uses a frequency/band-pass filter to isolate the fundamental frequency of the whistle in a permitted error range. This idea is comparable to the frequency mask in [3]. The second approach uses a logistic regression with l2-norm regularization. According to the paper, the second approach outperforms the first in all datasets.

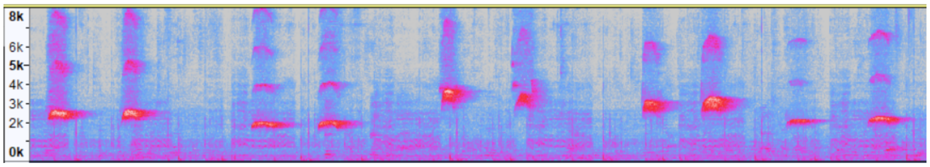

Fig. 1: Spectrogram of different whistles.

The SPL Team from Hamburg, the HULKs [9], are also using a frequency mask. However, instead of just filtering for the fundamental frequency, they also check if the first overtone exists. Since all whistles we examined had this clear pattern in the frequency domain, having at least one prominent overtone is a good indicator of a whistle. The spectrogram of different whistles is shown in Figure 1. The most dominant feature is the fundamental frequency, but also its multiple are visible.

The SPL team from Bremen, B-Human [1], uses a correlation to capture this unique pattern. Furthermore, to consider that different whistles have a different fundamental frequency, they have a sample of every whistle they tested. So their whistle recognizer can give the additional information which whistle was recognized.

In recent years neural networks are increasingly popular, so naturally, this approach deserves attention. For example, the HTWK Robots published their whistle recognizer [8], which uses a neural network after the Fourier transformation instead of a frequency mask. Another example of a neural network approach can be found in our code release from 2022 [12]. The main idea of using a neural network is to teach the network to recognize the characteristic whistle pattern shown in Figure 1. As can be seen in the evaluation, Section 4, this approach worked for us in most cases, but the neural network correlated sound volume with a whistle event. Since the human eye can easily spot this pattern of a whistle in the frequency domain, we did some research on adding prior knowledge to the system. Forssell and Lindskog's work about combining semi-physical and neural network modelling [7] inspired the new approach we present in this paper. Their method was explained by the example of a water tank system with the suggestion to modify it for sound in the future.

2 Dataset

For training a neural network, preparing the dataset is one of the most critical steps. Since the idea is to use the neural network as a pattern recognizer in the frequency domain, the audio samples must be prepared accordingly. This process involves buffering the audio stream to get windows containing 1024 samples. In the second step, we use a window function, the Hamming window, to avoid the Leakage-Effekt of a rectangular window. Following this, we compute the spectrogram of this window. Due to the symmetry of the Fourier transformation, we only need one half of the result plus the Nyquist frequency. After a Fourier transformation, the result contains complex numbers. However, we only use the real part of these complex numbers. For a more informative visualization, the volume/amplitude per frequency is shown in decibel (dB). This is not only justified by the concept of human hearing, but tests without the squared amplitude's logarithmic representation showed that the characteristic pattern, described in Section 1.1, was not as visible to the human eye.

In general, we use two different types of datasets, one to train, test and validate the neural network and one to tune parameters like the threshold for recognizing a whistle and benchmark how the whistle recognizer works in our framework. In both cases, we recorded the data with NAO Robots. For the first dataset, we extracted the audio data from log files. These log files were recorded during soccer games, tests and competitions, and explicit whistle tests. In addition, we cut off most redundant parts, like silence or robots walking, to have a slightly more balanced ratio between the events 'whistle' and 'no whistle'. However, we kept audio sequences which produced false positives. So in total, we have 369680 non-whistle and 18052 whistle samples after the adjustment, which makes a 20:1 ratio between the non-whistle and whistle samples.

The second dataset contains two log files. One was recorded during a show game at the Maker Faire in Rome. Again, due to the cheering crowd directly around the field, we noticed more false positives than in any other game, especially since at least one kid hit the fundamental frequency of the whistle. To further increase the difficulty, we used the log file of the goalkeeper, who was constantly near the spectators, instead of the field players. In addition, the referee, who holds the whistle, is moving around the field, so a whistle blown near the opponent's goal has the maximum distance. Unfortunately, real games contain fewer whistles compared to whistle tests. This log, in particular, contains only three whistles, a single whistle to indicate the start and two short whistles for the end of the game. The other log was a whistle test recorded in our lab in three steps. First, we only tested whistles where we control the airflow through the whistle. Since most teams and referees use handwhistles nowadays, the intensity and length of the sound highly depend on how fast and hard the currently whistling person is pressing on the air-filled ball attached to the whistle. Then, as a second step, we tested different types of noise to see what triggers false positives. And lastly, we combined both tests to get a more realistic benchmark where the desired sound is masked by noise. For the different types of noise, we went through a list of potential sounds caused by the robots, spectators, or other leagues. The list of sounds contains a robot falling and standing up in both directions, a whistle, a plastic bag for candy, finger-snapping, clapping, a human whistling, 'hit the pot', a circular saw, an angle grinder, a drilling machine, a vacuum cleaner robot and birds' twittering. In contrast to the first dataset, where we extracted the audio, we kept the second dataset as log files, which is an advantage and disadvantage at the same time. The argument for doing that was that we can replay these logs in our simulator and have the exact same conditions for every test. But unfortunately, our simulator is incompatible with other frameworks and extracting the audio to use loudspeakers would distort the result.

3 Whistle Recognition Architecture

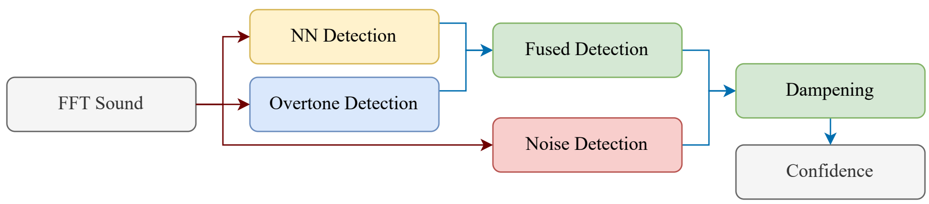

Fig. 2: Overview of our whistle recognition pipeline where red arrows indicate the flow of FFT information of the current audio and blue arrows indicate the confidence flow.

Our whistle recognition architecture is divided into three main parts, firstly a neural network, then a traditional method, and finally a damping factor that takes the ambient noise and volume into account. All the parts just mentioned generate a confidence value from an audio sample processed by a fast Fourier transformation (FFT). Figure 2 shows a simplified overview of the architecture and the dependencies of the modules we use in our whistle recognition pipeline. The complete recognition architecture runs in real-time on the NAO Robot.

3.1 Neural Network

In this section, we present our neural network approach in detail. It contains four convolution blocks of two types and a dense layer that merges the information and returns a confidence value, indicating whether a whistle is present. The first convolution block type consists of a one-dimensional convolution, a batch normalization layer, a leaky Rectified Linear Unit (ReLU), a one-dimensional max-pooling layer, and a dropout layer at the end. This combination should ensure that the neural network reacts less strongly to loud noises like our old approach, which is also visible in Section 4 and is thus more focused on the actual whistling patterns. The second convolution block type consists of only a one-dimensional convolution with an Exponential Linear Unit (ELU) as an activation function. Even without performing another computationally expensive batch normalization, this combination achieves a similar effect [5] as the previous block. Every convolution layer also uses a bias vector which is added to the output. The first three convolution blocks use a filter size of 32, 64, and 128 with a kernel size of 5 and a stride of two. After this, the second convolution block type uses a filter size of 64, a kernel size of 5, and a stride of one. The last stage of our neural network is a dense layer with one unit and a sigmoid function as an activation, which processes the flattened output of the second convolution block. As a result, the output of the last layer forms the confidence of our neural network c_{nn}. Furthermore, hyperparameters used for this architecture will be discussed in Section 4. Figure 3 shows a high-level perspective of the previously mentioned network design.

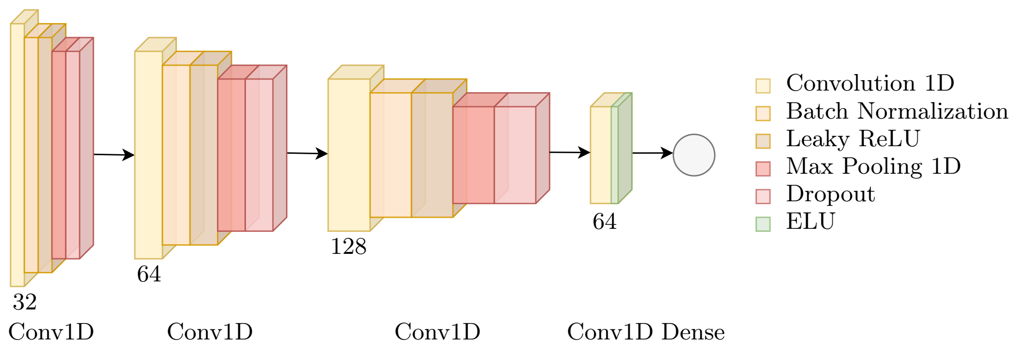

Fig. 3: Overview of our proposed whistle recognition neural network architecture where the numbers indicate the filter count of the convolution part of the current block and a step indicates that the input is reduced by half.

For the training of our neural network, we use the dataset described in Section 2, which is split into training- and validation data. Each dataset entry consists of 1024 samples of sound, which are Fourier transformed. This results in 513 amplitudes that are refined by $20 \cdot \log(amplitude)$ for further processing. Since our data still has a significant imbalance around 20:1 between background noise and real whistles, we further process our training data after splitting. Therefore we use the Imbalanced-learn Python library [10] to over-sample our training data with the synthetic minority over-sampling technique (SMOTE) [4] and then under-sample it with our variant of the edited nearest neighbors (ENN) algorithm [14]. We use the combination of SMOTE+ENN because, according to Batista et al. [2], this combination is well suited to balancing data for machine learning algorithms. This also gives better results in our empirical tests than using a loss specialized on imbalanced data like a focal loss [11]. The validation data used while training is not affected by this and still shows an imbalance typical for these data. To train our network with the training data generated in this way, we use the TensorFlow implementation of Rectified Adam (RAdam) as an optimizer in combination with the TensorFlow implementation of the lookahead mechanism; this combination is also referred as Ranger [15]. We then train our network for a maximum of 45 epochs with early stopping enabled to avoid overfitting because our neural network converges very fast.

3.2 Prior Knowledge Acquisition

Acquiring prior knowledge is a crucial part of our whistle recognition architecture. The acquisition is divided into two parts: a physical model-based detection of the whistle and a detection of the ambient volume and frequencies, which is treated as a noise indicator.

The model-based detection uses, just like the neural network, 1024 samples of sound, which are Fourier transformed as input. The resulting 513 amplitudes are then used to find a peak in a predefined range of frequencies. We define a peak as a whistle candidate if the amplitude of the peak is greater or equal to the average of the last 200 maximum amplitudes. After this, our model-based detection searches for the first overtone peak in the doubled frequencies range. If the amplitude of the overtone peak is greater or equal to the average amplitude of all samples, the whistle candidate is verified as a whistling sound.

Since all our detectors must output a confidence, a reasonable value must also be generated here. For this purpose, we use a gradient map that reflects the gradients between the amplitudes. Therefore, we extract the upward and downward gradient of the peak of the current whistle candidate. If the whistle candidate has been confirmed, the maximum absolute value of these two gradients, normalized by the current maximum amplitude, forms the confidence c_{pm}. Otherwise, the minimum absolute value of these two gradients, normalized by the maximum amplitude, forms the confidence. The ratio between the whistle candidate amplitude A and the current maximum amplitude A_{max} then weights the confidence. The strategy to compute the confidence of the model-based detection is shown in Equation 1.

$$c_{pm} = \begin{cases} \frac{A}{A_{max}} \cdot \max\left(\left|\frac{\delta_{max}}{A_{max}}\right|, \left|\frac{\delta_{min}}{A_{max}}\right|\right), & \text{if the whistle candidate} \\ & \text{is confirmed} \\ \frac{A}{A_{max}} \cdot \min\left(\left|\frac{\delta_{max}}{A_{max}}\right|, \left|\frac{\delta_{min}}{A_{max}}\right|\right), & \text{otherwise.} \end{cases} \tag{1}$$

Whistle recognition is especially difficult when a loud crowd and children are involved. Therefore, it is important to consider the volume and frequency that results from a crowd cheering or children. For this reason, we are introducing our noise detection. The problem is that the louder and larger the crowd that cheers is, the more frequencies become dominant. Therefore, we define our noise confidence as the ratio between the frequencies $\#f_c$ whose amplitudes exceeds a predefined limit and the number of all frequencies $\#f$, as shown in Equation 2.

$$\sigma = \frac{\#f_c}{\#f}. \tag{2}$$

In order to further improve the accuracy of the model-based whistle recognition, we have also added an automatic calibration of the frequency range in which we search for a peak. The automatic calibration is implemented as follows: each time a whistle is detected via the combined confidence, the corresponding whistle frequency, determined via the model-based whistle recognition, is stored in a buffer. That buffer holds the last ten detected whistle frequencies because this amount of detections is sufficient to compute an estimation of the current whistle frequency. The average f_{avg} of these frequencies is then used to determine the absolute deviation of the boundaries f_{min} and f_{max} of the frequency range, used to detect the whistle candidate. Due to the hardware and environmental fluctuation of the whistle frequency, it is mandatory to maintain a certain margin for the detection of the whistle. A margin of 250 Hz allows the whistle to be detected even if it deviates more strongly from the expected whistle frequency. Equation 3 shows the update strategy of our boundaries, where the 250 Hz grants a minimum distance between the expected whistle frequency and the boundaries.

$$\begin{aligned} f_{min} &= \min\left(f_{avg} - \frac{|f_{avg} - f_{min}|}{2}, f_{avg} - 250\ \text{Hz}\right) \\ f_{max} &= \max\left(f_{avg} + \frac{|f_{avg} - f_{max}|}{2}, f_{avg} + 250\ \text{Hz}\right). \end{aligned} \tag{3}$$

3.3 Combining Information

Combining all acquired information is an essential part of every ensemble. Our combination strategy is divided into two steps; the first one fuses the whistle recognition confidences, and the second one dampens the resulting confidence to consider the environmental conditions.

$$c_{nn/pm} = \frac{w_{nn} \cdot c_{nn} + w_{pm} \cdot c_{pm}}{w_{nn} + w_{pm}}. \tag{4}$$

We implement the fusion of our whistle confidences $c_{nn/pm}$ by a weighted average of the individual confidences. The weights w_{nn} and w_{pm} control whether the neural net or the physical model approach is more trustworthy, so prior knowledge is also introduced at this point. Equation 4 shows this process in detail.

$$c = c_{nn/pm} \cdot (1 - (w_{\sigma} \cdot \sigma)). \tag{5}$$

To ensure a trustworthy confidence even in different noisy environments, the fused confidence is attenuated by the sigma factor of Equation 2. For this purpose, the fused confidence is multiplied by the sigma factor, as shown in Equation 5.

4 Evaluation

The evaluation has several objectives: on the one hand, we compare the approach proposed here with different methods of other SPL teams and our old whistle recognition as an example of a neural network approach. On the other hand,

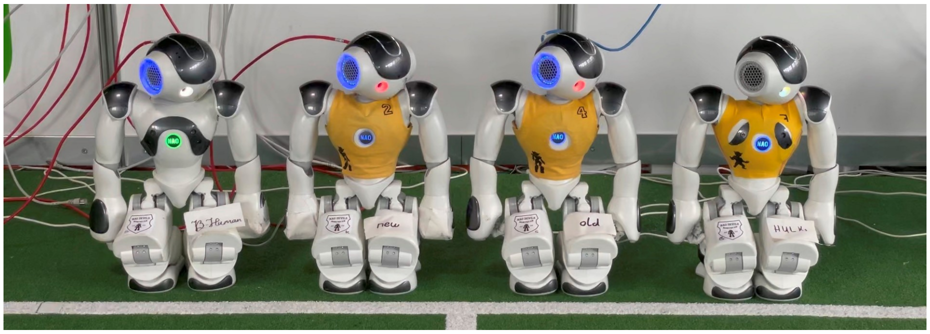

Fig. 4: Whistle comparison evaluation setup. Frameworks from left to right B-Human, the method presented here, our old approach and HULKs.

we show the impact the three presented main components have on the whistle recognition accuracy as well as their impact on the false positive rate.

As shown in Figure 4, we have chosen an experimental setup to compare our new approach with that of other SPL teams and our old approach. Therefore, each of the four approaches runs on a separate robot, and all robots sit close to the center line at the edge of the field. Afterwards, we disturb each robot (close) five to ten times per object with a cordless drill on different speed settings, a vacuum cleaner, and an angle grinder without whistling a whistle to see which approach falsely detected a whistle from this kind of noise. In addition, all robots are disturbed by selected noise objects from the other half of the field (far).

We then tested how well the different approaches could detect a whistle without additional noise. To do this, we blew a whistle ten times at different intensities, from soft to loud, and from various locations on an SPL robot soccer field in a quiet environment. Subsequently, we repeated this test five to ten times, again with different intensities of the airflow through the whistle, but this time masked with noise.

In addition, we performed an ablation study on a laboratory and an event dataset, described in Section 2. Here, we investigate the influence of the two detection components alone, how the combination of both components affects the result, and finally, what influence the inclusion of noise has.

For the first part, we use the False Positive Rate, i.e., how many interfering noises are falsely detected as whistles, and the True Positive Rate, i.e., how many whistles are detected despite interfering noises, as a metric, with the first one being slightly more important, as described in Section 1. For the second part, we use precision and recall as a metric to demonstrate the influence of our approach's different factors on the resulting score.

As shown in Table 1, the whistle recognition of B-Human is prone to most of our tested noise types. Our old neural network approach is vulnerable to loud noise from an angle grinder. In contrast, our approach presented here and HULKs whistle recognition is resistant against all noise types.

Table 1: A comparison of how different noise from different approaches is detected as a whistle (False Positive Rate). Lower is better.

	B-Human	HULKs	Old Approach	Our
Cordless Drill Level 1	0.2	0.0	0.0	0.0
Cordless Drill Level 2	0.2	0.0	0.0	0.0
Vacuum Cleaner (close)	0.6	0.0	0.0	0.0
Angle Grinder (far)	0.0	0.0	0.0	0.0
Angle Grinder (close)	1.0	0.0	0.8	0.0

Table 2: A comparison of how different noise affects the performance of varying whistle recognition frameworks (True Positive Rate). Higher is better.

	B-Human	HULKs	Old Approach	Our
None	0.6	0.9	0.9	**1.0**
Vacuum Cleaner (far)	0.7	0.7	**0.9**	**0.9**
Angle Grinder (far)	**1.0**	0.2	**1.0**	**1.0**

The effect of different types of noise on the True Positive Rate is shown in Table 2. As the table shows, most whistle recognition approaches work well in a quiet environment. The only exception to this is the whistle recognition of B-Human. However, this can probably be attributed to the fact that the B-Human framework is not able to handle weaker whistles well. Based on this baseline, the table shows that the framework of the HULKs has problems detecting a whistle as soon as noise is involved. However, the B-Human framework also has problems recognizing the whistle next to noise. This only changes significantly with the loudest noise, the angle grinder, because only then exclusively loud whistles whistled. Only our old approach, as well as our new approach, performed similarly well in this experiment. The second objective of our evaluation is to analyze the impact of the different components of our detection approach on the result.

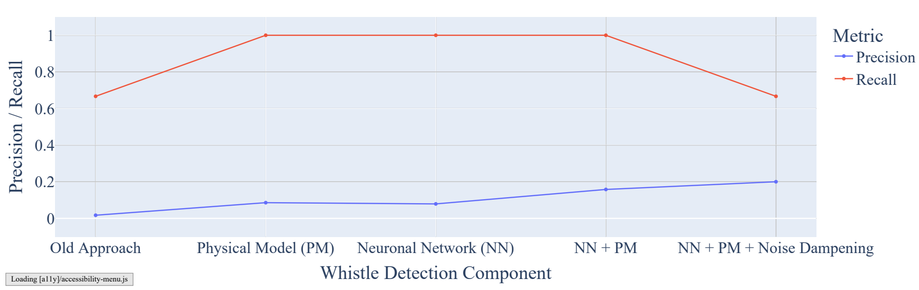

Fig. 5: Comparison of the influence of the different components of our approach on precision and recall during the Maker Faire Rome plus our old system since it proved to be promising in the previous tests.

Therefore, Figures 5 and 6 show, from left to right, the precision and recall of our old approach, the model-based detection alone, the neural network detection alone, our whistle recognition without the noise dampening component, and our whole whistle recognition architecture.

As already discussed in Section 2, the Maker Faire Rome dataset is quite tricky due to the cheering crowd directly around the field. The results of this dataset are shown in Figure 5, where our old approach has completely failed with a precision of practically zero. In contrast, our model-based method and our new neural network have a slightly higher precision on this dataset. The precision is further increased when the model-based method and the new neural network work together. The attenuation depending on the ambient noise increases the precision even further but decreases the recall.

Figure 6 shows that the precision and recall of our old approach is outperformed by every part of our new recognition method on the lab dataset, described in Section 2. Moreover, it is shown that the precision of our new neural network is higher than that of the model-based method, in contrast to the recall, which is higher for the model-based approach. Furthermore, it is visible that both methods together achieve a higher precision with only a slightly lower recall. When all three main components work together, the recall significantly increases while maintaining the same high precision.

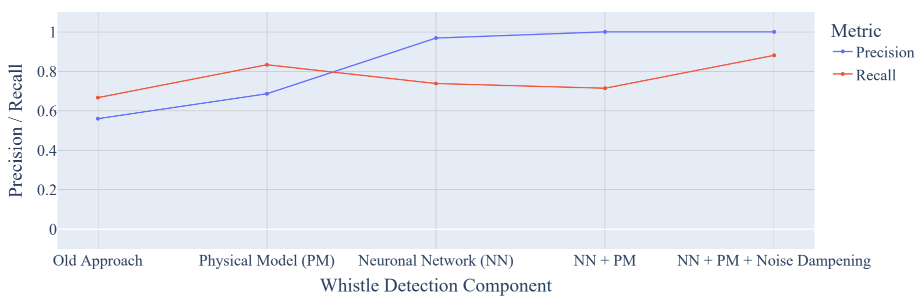

Fig. 6: Comparison of the influence of the different components of our whistle recognition on precision and recall under lab conditions plus our old system since it proved to be promising in the previous tests.

5 Conclusion and Future Work

The main goal of this work was to combine different strategies into one whistle recognizer to create a system that is more robust against various types of noise without missing whistle events. Additionally, runtime is a critical feature, especially on embedded hardware. As shown in Section 4, all tested approaches perform well under lab conditions. However, these conditions can't be guaranteed at all times. In addition, a varying airflow through the whistle and noise is challenging, especially in this combination. Nevertheless, we showed a significant improvement in noisy environments by combining the three described components while still creating an output in real-time. However, it would be nice to detect the third whistle on the data recorded in Rome as well, without decreasing the recall. A possible approach would be noise reduction/cancellation as a preprocessing. Another common error source are games on neighbouring fields when they are started around the same time. Right now, this difficulty is avoided by the organizers. In the future, however, it would be practical to determine the direction of the sound source.

References

1. B-Human Team: B-Human Code Release (10 2022), https://github.com/bhuman/BHumanCodeRelease/releases/tag/coderelease2022
2. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter **6**(1), 20–29 (2004)
3. Bonarini, A., Lavatelli, D., Matteucci, M.: A composite system for real-time robust whistle recognition. In: RoboCup 2005: Robot Soccer World Cup IX 9. pp. 130–141. Springer (2006)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
5. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus) (2016)
6. Committee, R.T.: Robocup standard platform league (nao) technical challenges pp. 2–7 (2023), https://spl.robocup.org/wp-content/uploads/SPL-Challenges-2023.pdf
7. Forssell, U., Lindskog, P.: Combining semi-physical and neural network modeling: An example of its usefulness. IFAC Proceedings Volumes **30**(11), 767–770 (1997)
8. HTWK Team: HTWK Whistle Detection (02 2022), https://github.com/NaoHTWK/WhistleDetection
9. HULKs Team: HULKs Code Release (12 2022), https://github.com/HULKs/hulk/releases/tag/coderelease2021
10. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research **18**(1), 559–563 (2017)
11. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
12. Nao Devils Team: Nao Devils Code Release (02 2022), https://github.com/NaoDevils/CodeRelease/releases/tag/CodeRelease2022
13. Poore, K., Abeyruwan, S., Seekircher, A., Visser, U.: Single-and multi-channel whistle recognition with nao robots. In: RoboCup 2014: Robot World Cup XVIII 18. pp. 245–257. Springer (2015)
14. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics (3), 408–421 (1972)

15. Wright, L.: Ranger - a synergistic optimizer. https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer (2019)