IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.16, No.3, July 2022, pp. 261~270
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.75697

Spectrogram Window Comparison: Cough Sound Recognition using Convolutional Neural Network

Dzikri Rahadian Fudholi*[1], Novia Arum Sari[2], Muhammad Auzan[3]
[1,2,3]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]dzikri.r.f@ugm.ac.id**, [2]noviaarums@gmail.com, [3]muhammadauzan@ugm.ac.id

Abstrak

Batuk adalah sebuah gejala penyakit paling umum terutama untuk penyakit pernafasan. Deteksi batuk secara cepat menjadi kunci terutama pada kondisi terkini yaitu COVID-19. Pengenalan batuk yang baik adalah yang menggunakan alat yang tidak mengganggu manusia dengan menempel sensor yaitu dengan suara saja. Untuk melakukan deteksi suara ini, digunakan metode Deep Learning yang paling memiliki hasil paling bagus yaitu Convolutional Neural Network (CNN). Namun, CNN membutuhkan masukan data dalam bentuk citra, sedangkan suara berbentuk satu dimensi. Sebuah tambahan proses perlu dilakukan yaitu mengubah data suara menjadi data citra yaitu dalam bentuk spectrogram. Namun dalam pembentukan spectrogram muncul pertanyaan, berapa ukuran spectrogram yang akan menghasilkan hasil paling baik. Penelitian ini membandingkan ukuran spectorgram mana yang paling baik dari sisi presisi dan recall. Hasil yang didapat adalah data spectrogram dengan jendela 4 detik memiliki nilai performa F1-score paling tinggi yaitu 92.9%. Sehingga untuk melakukan pengenalan jenis suara, didapatkan window sekitar 4 detik itu lah yang akan menghasilkan performa tinggi.

Kata kunci*—Spectrogram Window, Convolutional Neural Network, Cough Sound, Deep Learning*

Abstract

Cough is one of the most common symptoms of diseases, especially respiratory diseases. Quick cough detection can be the key to the current pandemic of COVID-19. Good cough recognition is the one that uses non-intrusive tools such as a mobile phone microphone that does not disable human activities like stick sensors. To do sound-only detection, Deep Learning current best method Convolutional Neural Network (CNN) is used. However, CNN needs image input while sound input differs (one dimension rather than two). An extra process is needed, converting sound data to image data using a spectrogram. When building a spectrogram, there is a question about the best size. This research will compare the spectrogram's size, called Spectrogram Window, by the performance. The result is that windows with 4 seconds have the highest F1-score performance at 92.9%. Therefore, a window of around 4 seconds will perform better for sound recognition problems.

Keywords— *Spectrogram Window, Convolutional Neural Network, Cough Sound, Deep Learning*

Received June 24th,2022; Revised July 15th, 2022; Accepted July 18th, 2022

1. INTRODUCTION

Cough is a widely occurring symptom that can happen to all humans; hence it is needed to do recognition for monitoring purposes. The cough is a symptom of over one hundred medical conditions, including respiratory diseases [1]. During the COVID-19 pandemic, cough can be a decisive factor in recognizing diseases. The application of a cough sound recognition can vary from cough detection in a particular room for warning to cough counting for each person. Therefore, there is the need to do cough sound recognition, especially with just using the sound.

Although the detection of cough may not only be from sound, it is the most noticeable. There are other signs of people coughing, such as closing their mouth with their hands and moving their body forward and backward as they cough. However, those signs are subtle if compared with the sound it makes. Sound is vital in differentiating as it can be used to differentiate the rightness of pronouncing words [2]. In addition, sound receptors such as microphones are omnidirectional, and they can hear a whole 360° where the maximum of a camera to record an image without too much distortion is around 120°. It can be said that sound is more convenient to do recognition, but identifying through sound may be more complex.

A sound-only cough recognition is better in various aspects. More than one tool is used for cough recognition, namely mic, sensor array, wearable, and smartphone [3]. Each has its sensitivity rate and uses a different algorithm. However, [3] chooses a smartphone's built-in mic to do the recognition. Using a smartphone mic means that the only data used is sound. This option is much cheaper than using an embedded sensor in a wearable that produces more than just sound data. In addition, sound-only data with a device not attached to a person's body is non-intrusive; hence, it is easy to use. In [3], previous sound experiments only produced a better result.

The machine learning approach has been used nearly everywhere and can achieve great things. A classic machine learning algorithm such as Support Vector Machine can do music classification [4]. A more modern approach using a Deep Learning algorithm that uses an image called Convolutional Neural Network (CNN) was able to do aspect-based sentiment analysis [5]. The CNN method is the better algorithm among the Deep Learning methods, especially in image datasets. Another Deep Learning approach, a Recurrent Neural Network (RNN), excels in a sequenced dataset. However, in cough sound detection, CNN performs better than RNN [6]. What makes CNN better is its ability to capture essential features in an image. In the end, it is commonly known that when deep learning is said, it means CNN.

Sound data is sequenced, and its shape is a one-dimension array. Its original shape does not fit the CNN algorithm; hence we need to change it. The spectrogram is one algorithm to change a sound signal into an image signal in a two-dimensional array [7]. The need for two-dimensional data means that in every sound recognition task, it is needed to convert the sound file into a spectrogram. However, there is a limit to converting a continuous dimension array into a discrete two-dimension array. There will be data loss if the data is cut and not all data sequences can fit in a single image or call a single window.

Many possible data windows exist while using a sound dataset for a particular algorithm. A window is a set of individual data that creates a certain length, and it is constant along with the usage of the model. The length can be in total data or, in the sound case, seconds. Some research chooses the window without explaining why; for example, [8] chooses a one-second window, and [1] chooses a maximum of hundreds of milliseconds. In addition, [9] and [6] minimally explain their chosen window details. The journey to find the best window for cough sound recognition is not easy. There are many variables in play; one of them is a contradiction of the need, fast vs precision which is small window vs large window respectively.

Therefore, this research will compare and find a better window among the many possibilities, especially for the cough sound dataset using a deep learning algorithm.

2. METHODS

This research aims to find a better window size for sound recognition problems, especially cough sounds. Each window of choice is the same: choosing the dataset, preprocessing, creating the model and calculating the performance. In this research, the window size is the length in milliseconds of the sound regardless of the sampling on the sound dataset. However, the dataset used for the comparison used a 48kHz sampling frequency.

The methods of this research consist of a few steps, as seen in Figure 1 where the research method starts with splitting the Coswara dataset into windows. Then it converted the sound data to spectrogram as the data preprocessing. After that, the spectrogram data is fed to CNN for training, and the performance is measured and compared.

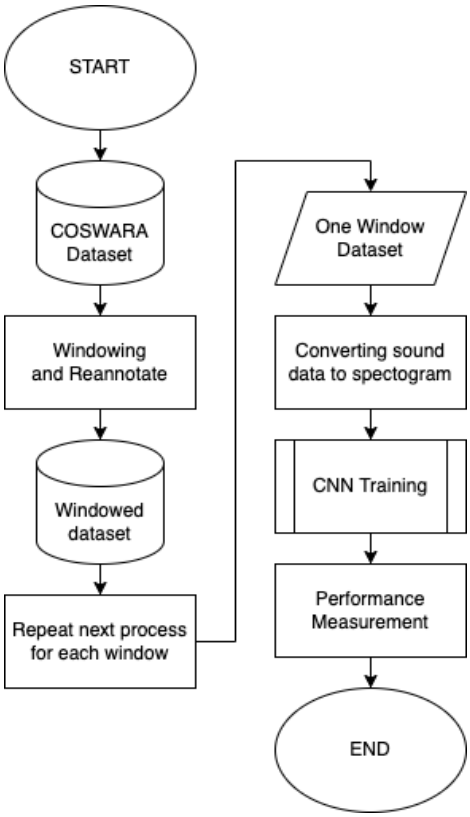

Figure 1. Flowchart of the whole method

2.1 Dataset

With the current pandemic situation with cough as the primary symptom, there are many available datasets online. The Coughvid [10] and Coswara [11] are the most recent cough dataset. Both records cough sound to classify as a covid patient or not. This research will use the Coswara dataset, containing cough, breathing and speaking sounds. The advantage of using a newer dataset is that sound quality is better than before and represents current technological advancement. Therefore this research will apply in the future.

The Coswara dataset [11] has a participant count of 941, mainly from India (about 849). Most participants are Male, with 719 people and 222 Female, 780 healthy and 104 unhealthy,

ranging in ages from around 10 to 70 years old. Each participant was asked to make nine sounds, including cough and shallow and heavy breath. The audio sampling of Coswara is 48kHz, and each sound duration is from one to five seconds. Finally, each sound from the participant was annotated by 13 annotators.

There are many variables inside the Coswara dataset [11], there are many variables, as seen in Figure 2. Coswara dataset count. The first variable is the words or the activity in which the sound is recorded. The activities are: 1. enunciate vowel o for a few seconds, 2. enunciate vowel e for a few seconds, 3. enunciate vowel a for a few seconds, 4. shallow coughing or coughing lightly, 5. heavy coughing or coughing hardly, 6. shallow breathing or slow and light, 7. deep breathing or heavily, 8. counting the number from zero to twenty with average speed, 9. counting number from zero to twenty with faster speed. Of these 9 activities in the dataset, the ones considered as the cough are only numbers 4 and 5, both shallow and heavy. The other 7 activities are considered non-cough sound. This activity is the one that is being used in this research. The second is the sound quality, which has 3 levels: clean, noisy, and bad. However, in this research, we disregard the sound quality as the aim is to differentiate cough sound or not.

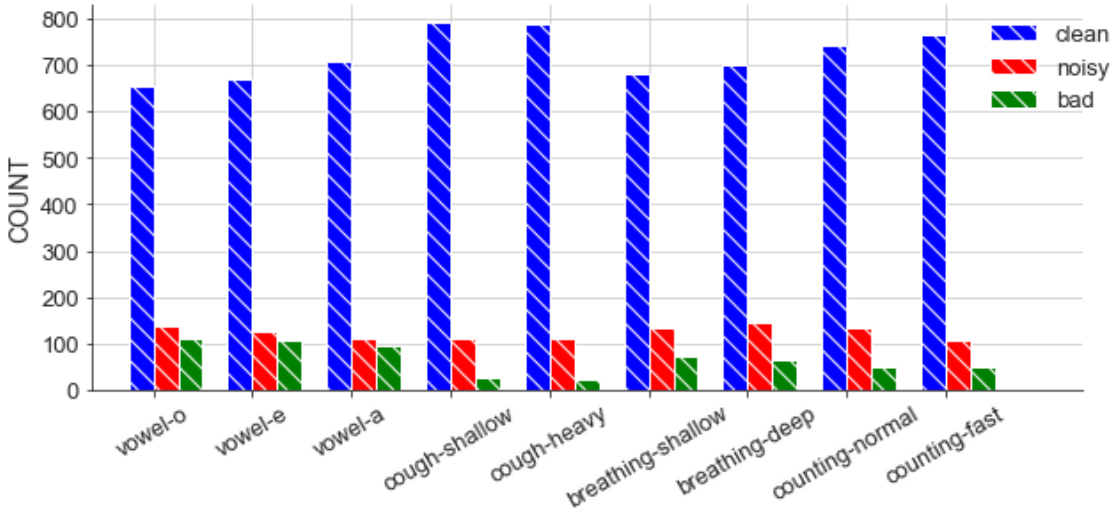

Figure 2. Coswara dataset count

Enhancing the Coswara dataset, additional annotation is made to accommodate the windowing. Each window from the cough sound is reannotated for each window because there is a possibility that a part of the cough-classified sound does not have an actual cough sound. Therefore, windowing is in dataset preparation rather than splitting in the model learning. This research will cover ten windows which is: 50ms; 100ms; 200ms; 400ms; 800ms; 1s; 2s; 3s; 4s; 5s. Each of that windows will be produced from the Coswara dataset and reannotated as cough sound and non-cough sound.

Splitting the dataset into the ten windows consists of splitting and reannotating. Splitting the dataset follows the N-Gram technique in [12], where each millisecond is the gram, and the N is the length. For example, a single 110ms will produce 11 windows of 100ms sound data. After that, each window is reannotated with the same annotation as its parent. For example, if the previous single 110ms sound has a cough class, then all 11 windows of 100ms sound will also have the cough class. It produces more datasets than cutting the sound file into the desired window. The smaller the window, the more the total sound data in the dataset.

2.2 Preprocessing

CNN input is two-dimensional in the form of a matrix. Meanwhile, sound data in .wav data is one-dimensional data. Therefore, to use sound data in the .wav file type, changing the

one-dimension data into two is needed. In [8], spectrogram was chosen as the preprocessing step as it achieves excellent performance in cough recognition. In this experiment, we will do the spectrogram using Python with the help of the Tensorflow [13] library, similar to the Librosa [14] library but done manually.

2.3 Model Training

As proposed in the introduction, this research will use CNN for the model. The CNN in this research is using Tensorflow [13]. In Tensorflow, they provide a lite version that applies the trained model to a much smaller, less powerful device to do the recognition, hence easily applied to current smartphones. This research also used SciKit Learn [15] library to assist the training process. Finally, the library chosen for the model is from Keras [16] library, which is the extension of Tensorflow [13].

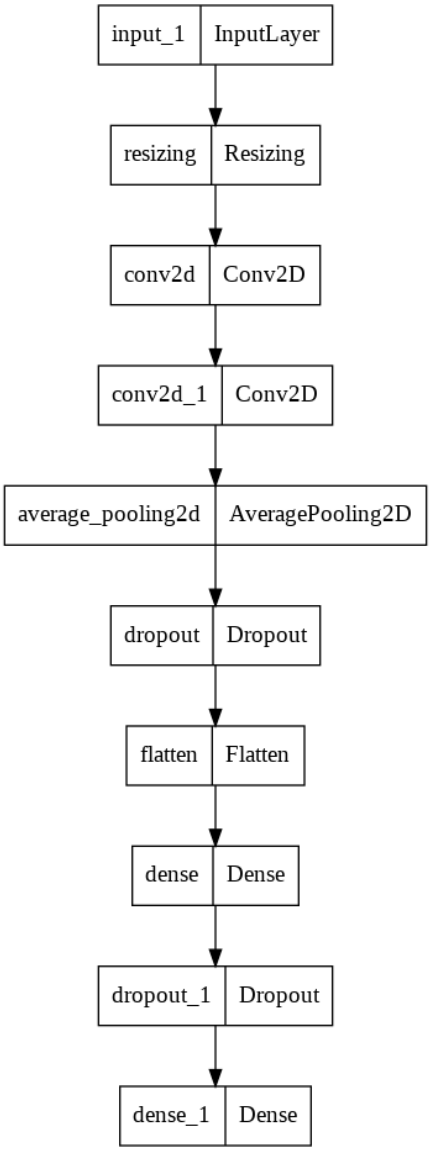

Figure 3. CNN architecture

Building the model has some configuration to choose from, an almost infinite option. As seen in Figure 3, the model starts with an input layer. After the input is in the resizing layer, the input is converted to a 64x64 matrix. Following are two convolution layers with every 32

and 64 filters, with 3 kernels and Relu activation. After the convolutions layer, an average pooling is applied, followed by dropout with the rate of 25%. Then the data is flattened and input to a dense layer with 128 nodes with Relu activation. Finally, the second dropout is used with a 50% rate and the last output layer of 1 node with sigmoid activation. The last layer will output the probability of whether the sound input is considered a cough or non-cough

2.4 Performance Measurement

In order to evaluate the performance of the model, we will use the commonly used performance measurement for CNN, which is Accuracy, Precision, Recall, and F1-measure [17]. Each of those performances calculates using these four values:

1. True Positive (TP): predicted as 1 and is 1 in the real dataset,
2. True Negative (TN): predicted as 0 and is 0 in the real dataset,
3. False Positive (FP): predicted as 1, but the real one is 0,
4. False Negative (FN): predicted as 0, but the real one is 1.

The number 1 above is the correct class, and 0 is the incorrect class. For example, in this research, 1 is considered a cough and 0 is considered a non-cough sound.

Accuracy calculates using all four values where the sum of TP and TN is divided by the sum of all four. It indicates how accurate the model predicts correctly whether the 1 or 0 class. However, accuracy is quite obsolete as it also uses TN, where some classification problems did not care because they care more about how well the correct class is to be predicted, not otherwise.

Precision is calculated by dividing TP by the sum of TP and FP. Just dividing TP with all the correct (all positive) predictions means that precision is concerned with the ability of the model to predict the correct class. On the other side of precision is recall which is calculated by dividing TP by the sum of TP and FN. Because recall uses the division on TP and FN, the score is calculated by comparing to all real correct classes. This explanation means recall is calculating how many correct classes does the model correctly predicting.

Precision and recall are similar yet different. Both have different functions, and both are used for different occasions. Precisions are used if having less FP is more critical such as predicting in credit scoring where the correct class is the person allowed to take credit. While the recall is the opposite, where having less FN is more critical. For example, predicting Covid is better for people to be predicted as positive rather than negative. However, there are times when both are equally important, hence the F1 score. The number 1 in the F1 score means that the importance of precision and recall is 1 by 1, equal. F1-score is calculated by dividing precision times recall times 2 by the sum of precision and recall. The number 2 can be changed to 1 + beta, where the beta is the weighting between recall and precision. In this research, precision and recall are considered equal in importance; hence the final performance score will use the F1 score.

3. RESULTS AND DISCUSSION

There are few results from the cough sound data recognition using CNN. The results are a comparison chart in the accuracy, precision, recall and F1 score. All the presented comparison charts consisted of the score and the window. The score, which is the height, has a minimum number of 0%, and the maximum is 100%. Generally, the more the score, which is close to 100%, is considered good; otherwise, 0% is considered flawed. Meanwhile, the window consists of ten windows starting with 50ms from the far left to 5000ms to the far right. This window number or length from 50 to 5000 in Figure 4 to Figure 7 is the milliseconds of the data used for the classification model.

Starting with the accuracy score in the result, as is seen in Figure 4, there is a trend of increase starting from 50 ms at 73.46% to the peak at 4000ms at 93.05%. The increase does not

happen for 5000ms, which decreases slightly to 91.7%. From Figure 4, the significant improvement is in 3000ms, with a 4% increase from 2000ms and just a 1% difference to 4000ms. From these accuracy scores, it can be said that windows around 3000ms to 5000ms have enough information to distinguish between cough sound and not because there was where the peaks are.

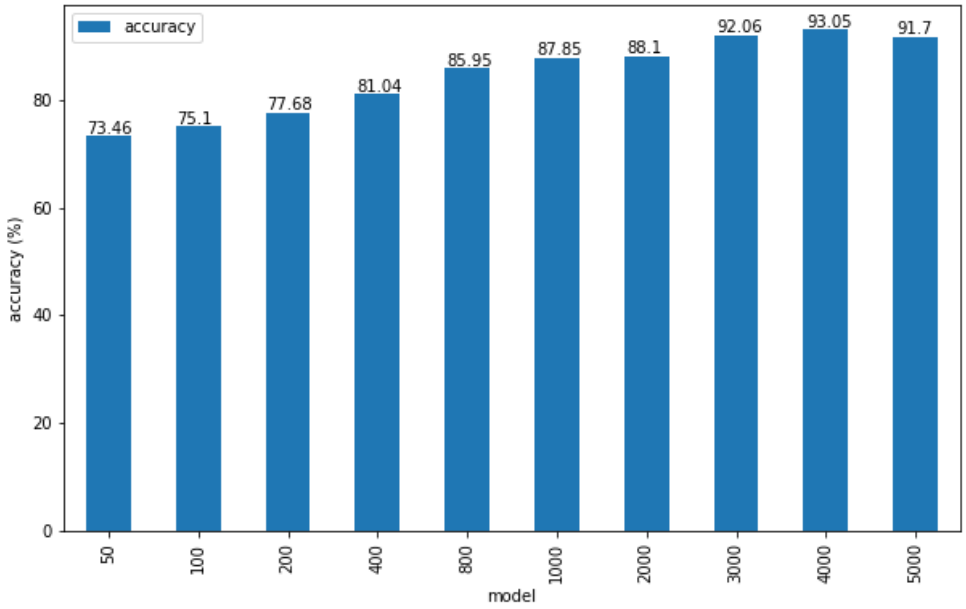

Figure 4. Accuracy comparison

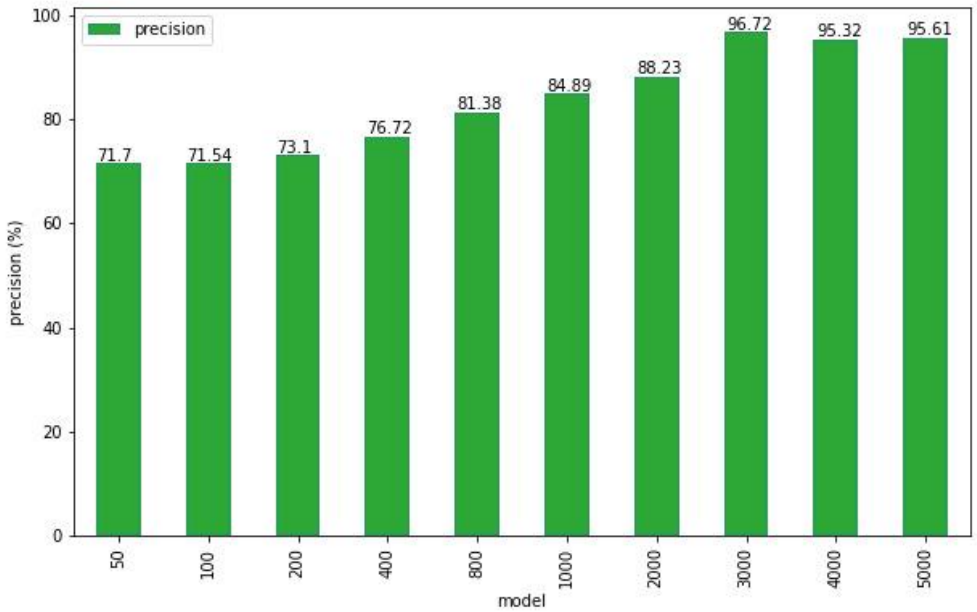

Figure 5. Precision comparison

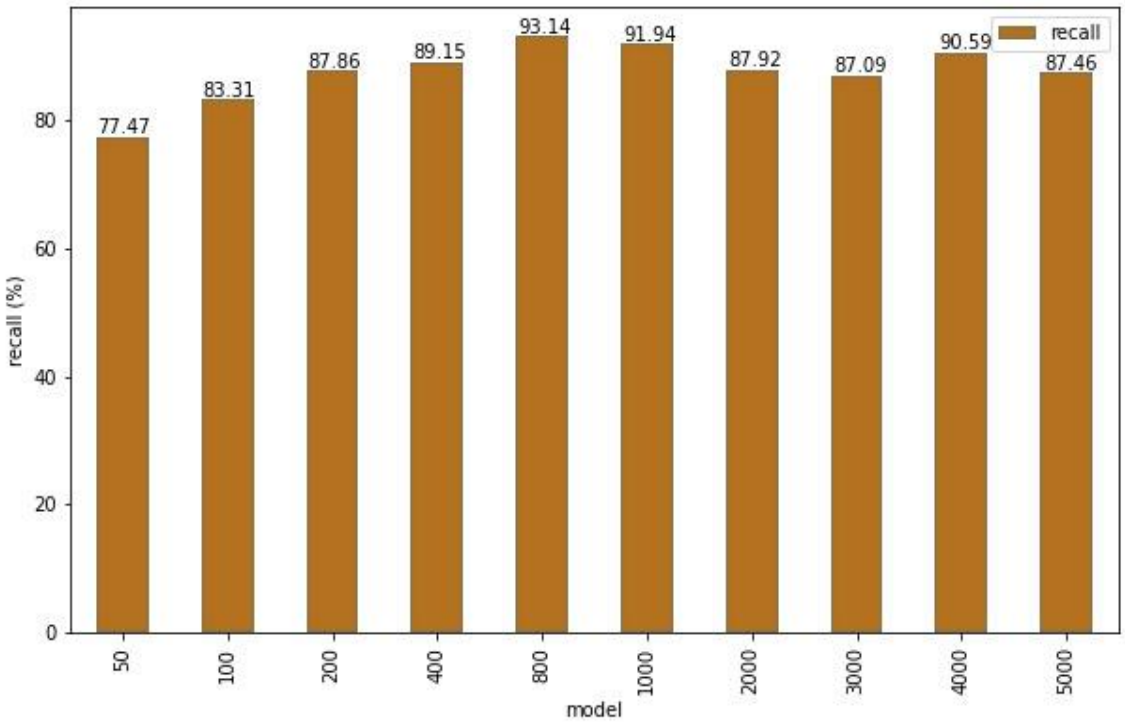

Figure 6. Recall comparison

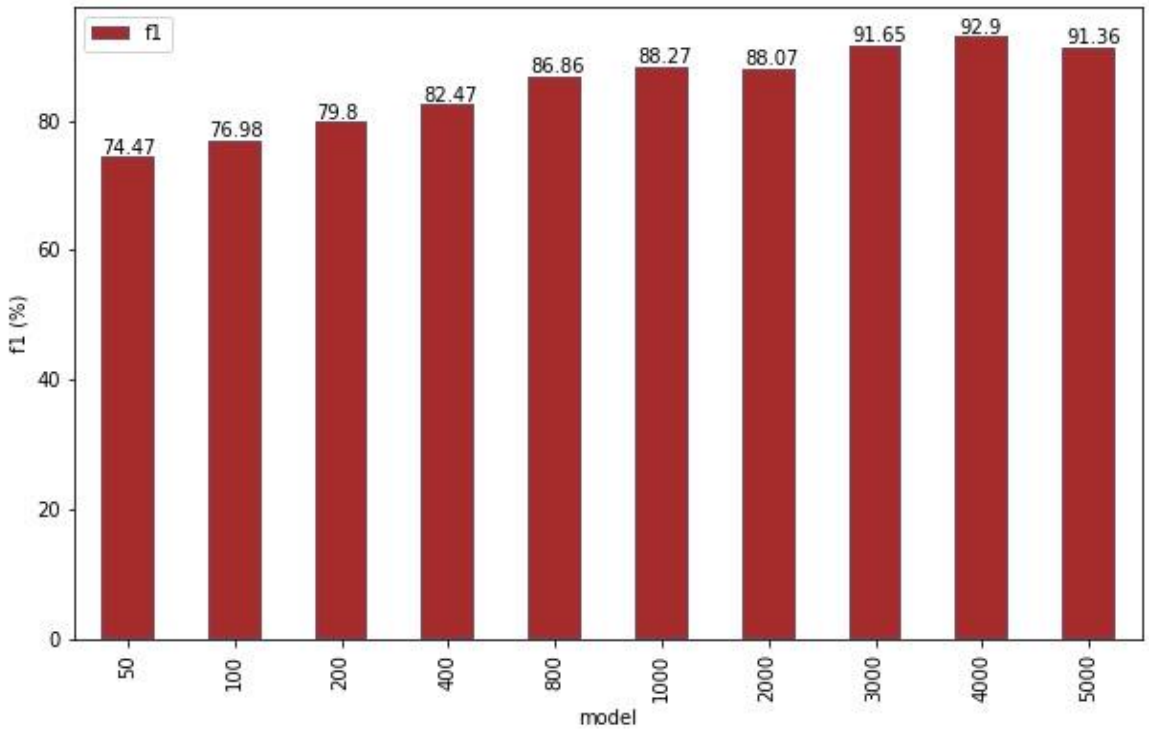

Figure 7. F1-Score comparison

For the precision, it can be seen in Figure 5 where the precision score fluctuates a little at 4000 ms, and the peaks are in 3000ms at 96.72%. From 2000ms backwards, the precision score reduces and converges at around 72% for 200ms, 100ms, and 50ms. The overall score is very similar to accuracy scores as both accuracy and precision measure similar peak groups of 3000ms to 5000ms.

For recall comparison, there is wider fluctuation, as seen in Figure 6. Interestingly, the fluctuations are even from 5000ms to 200ms, where the peak is 800ms with 93.14%. Then it was followed by a little decrease to 77.47% at 50ms window. From the recall result, we can say that using around a 400ms window is enough to do sound recognition and can differentiate between a cough and a non-cough.

Lastly, the F1 score combines precision and recall with the same weight. Figure 7 shows that the F1-measure falls similar to accuracy, with an increase from 50ms to the peak at 4000ms and a reduction at the final window of 5000ms. The peak 4000ms is 92.9%, and the slump of 50ms is 74.46%. The trend is similar to accuracy because the F1 measure considers precision and recall equally.

4. CONCLUSIONS

There is a dilemma between real-time data, which is using data as small as possible to get the result as soon as possible or finding the best result using more complex data. Considering the final F1 score, there is a good fact that a small data window under 400ms is not enough to get a good result which is more than 80%. Considering a higher threshold of 90%, having a data window of more than 3000ms and less than 5000ms is preferred. Finally, there is no ideal window size. However, for a better sound recognition problem, a size between half a second to four seconds is considered acceptable. The choice will depend on how much buffer is wanted for the system to run; four seconds may be too long.

ACKNOWLEDGEMENTS

This research is funded by *Dana Masyarakat FMIPA UGM*. This research is supported by Tarangga, Nurrizki, Azhara, Vincent, and Norman as part of the sound research group.

REFERENCES

[1] J. Monge-Alvarez, C. Hoyos-Barceló, K. Dahal, and P. Casaseca-de-la-Higuera, “Audio-cough event detection based on moment theory,” *Appl. Acoust.*, vol. 135, pp. 124–135, 2018.

[2] D. Fudholi and H. Suominen, "The Importance of Recommender and Feedback Features in a Pronunciation Learning Aid," 2019, pp. 83–87, doi: 10.18653/v1/w18-3711.

[3] F. Barata, K. Kipfer, M. Weber, P. Tinschert, E. Fleisch, and T. Kowatsch, "Towards device-agnostic mobile cough detection with convolutional neural networks," in *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, 2019, pp. 1–11.

[4] B. R. Ismanto, T. M. Kusuma, and D. Anggraini, "Indonesian Music Classification on Folk and Dangdut Genre Based on Rolloff Spectral Feature Using Support Vector Machine (SVM) Algorithm," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 1, pp. 11–20.

[5] P. R. Amalia, "Aspect-Based Sentiment Analysis on Indonesian Restaurant Review Using a Combination of Convolutional Neural Network and Contextualized Word Embedding," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 3.

[6] S. A. H. Tabatabaei, G. Augustinov, V. Gross, K. Sohrabi, P. Fischer, and U. Koehler, "Automatic Detection and Classification of Cough Events Based on Deep Learning," *Curr. Dir. Biomed. Eng.*, vol. 6, no. 3, pp. 322–325, 2020.
[7] J. Monge-Álvarez, C. Hoyos-Barceló, L. M. San-José-Revuelta, and P. Casaseca-de-la-Higuera, "A machine hearing system for robust cough detection based on a high-level representation of band-specific audio features," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 8, pp. 2319–2330, 2018.
[8] Q. Zhou *et al.*, "Cough Recognition Based on Mel-Spectrogram and Convolutional Neural Network," *Front. Robot. AI*, vol. 8, 2021.
[9] S. Matos, S. S. Birring, I. D. Pavord, and H. Evans, "Detection of cough signals in continuous audio recordings using hidden Markov models," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1078–1083, 2006.
[10] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms," *Sci. Data*, vol. 8, no. 1, pp. 1–10, 2021.
[11] N. Sharma *et al.*, "Coswara--A Database of Breathing, Cough, and Voice Sounds for COVID-19 Diagnosis," *arXiv Prepr. arXiv2005.10548*, 2020.
[12] A. R. Isnain, N. S. Marga, and D. Alita, "Sentiment Analysis Of Government Policy On Corona Case Using Naive Bayes Algorithm," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 1, pp. 55–64, 2021.
[13] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.
[14] B. McFee *et al.*, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in science conference*, 2015, vol. 8, pp. 18–25.
[15] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[16] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.
[17] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv Prepr. arXiv2010.16061*, 2020.