Performance Improvement Using CNN for Sentiment Analysis

Moch. Ari Nasichuddin[1], Teguh Bharata Adji[2], Widyawan[3]

***Abstract*—The approach using Deep Learning method provides great results in various field implementations, especially in the field of Sentiment Analysis. One of Deep Learning methods is CNN which has the ability to provide great accuracy in some previous research. However, there are some parts of the training process which can be improved to upgrade the accuracy level and the training time. In this paper, we try to improve the accuracy and processing time of sentiment analysis using CNN model. By tuning the filter size, frameworks, and pre-training, the results show that the use of smaller filter size and pre-training word2vec provide greater accuracy than some previous studies.**

***Keywords*— CNN, Deep Learning, Sentiment Analysis.**

I. INTRODUCTION

The Big Data era provides the impact of abundant data available on the internet, especially on the text data. To get information from the data, we need to analyze the text by using an appropriate method, for example, sentiment analysis. Some machine learning methods can be used in sentiment analysis cases. For example, Neural Network (NN), a method that imitates the working of biological neural networks. The basic component of NN is a neuron, it serves as a quantifier and non-linear mapping processor. Between one neuron with another neuron is connected by a value called weight [1], [2].

There are several neurons or nodes in each layer. Basically, there are three major layers in NN consisting of the input layer, hidden layer, and output layer, as (see Fig. 1). The input layer is the layer of incoming data which will be continued to the next layer. Hidden layer transforms the inputs into something that the output layer can be used as input. This layer is an important position in processing complex problems. Output layer is a layer for the result of the input value from previous processes [3].

Deep Learning is a kind of NN, but Deep Learning has more hidden layers than common NN (see Fig. 2), so that Deep Learning has an ability in accomplishing complex problems. Deep Learning has several variants including Convolutional Neural Network, Recurrent Neural Network, and Recursive Neural Network.

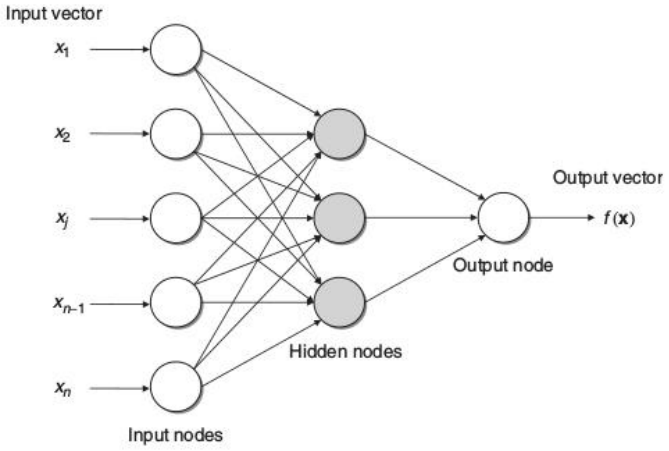

Fig. 1 Neural Network architecture.

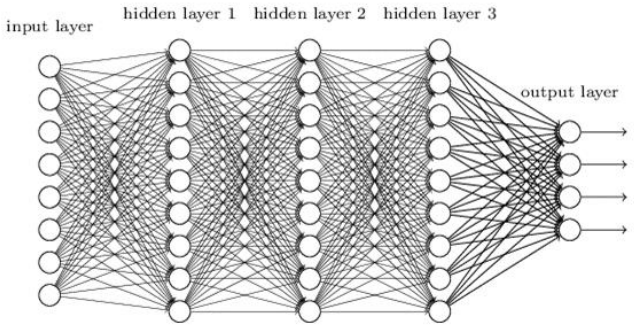

Fig. 2 Deep Learning architecture.

The implementation of Deep Learning in sentiment analysis cases has been studied by some researchers, such as a study that attempted to apply a single layer of convolution CNN with a bit of tuning to its hyperparameter and word2vec as pre-training [4]. The data set used in this study is Movie Review and Twitter. The result of the Movie Review data set produced 76.1% accuracy without utilizing word2vec and 81.0% by utilizing word2vec. And on Twitter data set produced 82.7% accuracy without utilizing word2vec and 86.8% by utilizing word2vec.

CNN was also used for a sentiment analysis. The results showed the use of pre-training word2vec, filter size region, and the number of feature maps achieved 81.65% accuracy [5]. Another study tried to combine CNN with LSTM (ConvLSTM) for the training process. By using the Movie Review data set, the research showed an accuracy up to 88.3%. These results indicated the use of pre-training had an effect on accuracy [6].

[1] *Master Student, Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta 55281 Indonesia (e-mail: ari.n@mail.ugm.ac.id)*

[2,3] *Co-Author, Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta 55281 Indonesia (e-mail: adji@ugm.ac.id, widyawan@ugm.ac.id)*

A CNN training without using pre-training word2vec but using pre-training One Hot Encoding has been studied. The experiment was performed by tuning the hypermeters used. The results showed the use of a smaller filter size gave an accuracy up to 77.4%. The use of a smaller filter size gave a positive impact on increasing accuracy and training time [7].

A CNN experiment with three layers of convolution and added pre-training word2vec. With more complicated architecture it only got 45.4% accuracy [8].

Based on the studies above, the main factors in CNN tuning are on the use of filter size and pre-training. Therefore, this proposed method aims to improve performance by tuning the use of filter size, pre-training, and add variations of the framework used.

II. METHOD

The CNN architecture used in this proposed study has three main layers, the Input Layer, the Feature-extraction Layer, and the Classification Layers (see Fig. 3). The first layer is the Input Layer. In this layer, there is a Word Representations Matrix that saves the pre-training results sentence. The pre-training process can use one hot encoding and word2vec. The example for one hot encoding can be seen in Fig. 4.

As seen in Fig. 4, an example sentence "I Like This Movie Very Much!" consists of seven words, so 7 x 7 Matrix Word Representations will be generated. Each word of the sentence is mapped into the diagonal elements of the matrix. All the elements/cells in diagonal where the position of the word is located will be given the value of 1 (one), while the other elements/cells not containing the word are given a value of 0 (zero). For example, the word "I" is in the cell located in the first row of the first column of the matrix, then the cell should be at 1, while the other cells in the first row have been valued at 0. Therefore, in the first row of the matrix should be valued at "1 0 0 0 0 0 0 ". Continued on the second row of the second column, the word "Like" is the second word in the sentence, so the value 1 will be given to the cell located on the second row of the second column and the other cells on the second row are valued at 0. So, the second row matrix generates values "0 1 0 0 0 0 0". That method must be applied to the following columns, i.e., the value 1 is given to the cell located in the third row of the third column for the third word, the fourth row of the fourth column for the fourth word, the fifth row of the fifth column for the fifth word, the sixth row of the sixth column for the sixth word, and the seventh row of the seventh column for the seventh word. The other cells not mentioned are valued at 0 because those cells are not containing any words.

After the Word Representations Matrix has been scored by one hot encoding, the next process is in the Feature-extraction Layer. The process on this layer will be started when the total Filter Matrix is 6 (2 2x7 Filter Matrix, 2 3x7 Filter Matrix, 2 4x7 Filter Matrix) (see Fig. 3(b)) and sliding over the Word Representations Matrix (see Fig. 3(a)). The number of each Filter Matrix used is 2 to capture more information. In order to generate an initial value, each Filter Matrix should contain a randomly-initialized number.

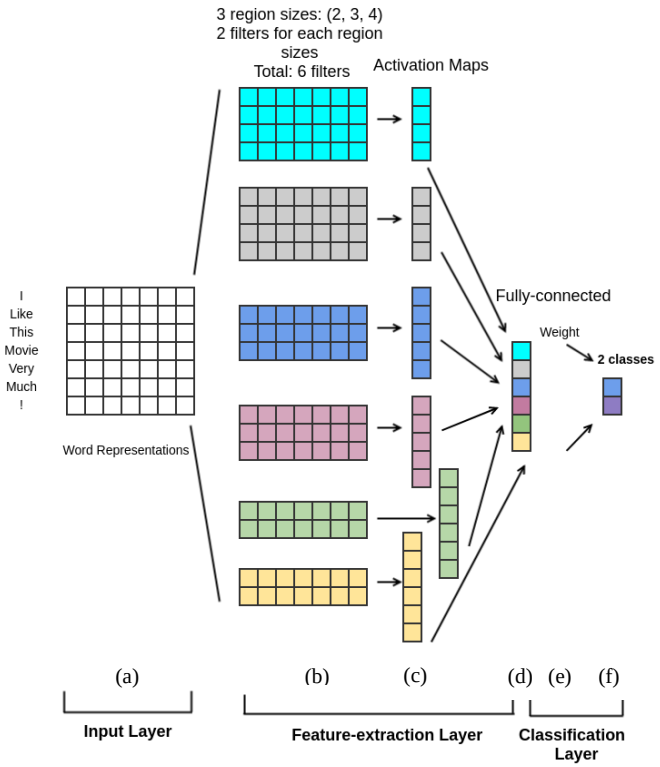

Fig. 3 CNN architecture.

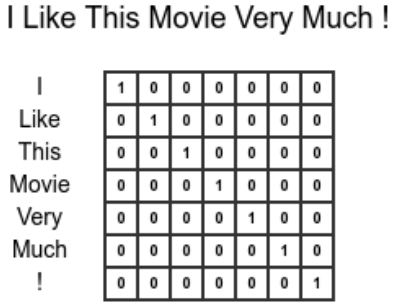

Fig. 4 Result of one hot encoding process.

The following example uses a 2x7 Filter Matrix (see Fig. 3(b) of the yellow Filter Matrix). The 2x7 Filter Matrix on the left of the Word Representations Matrix does the first sliding on the first and second rows of the Word Representations Matrix. The 2x7 Filter Matrix to the right of the Word Representations Matrix does the second sliding on the second and the third rows, etc (see Fig. 5). In each sliding process, there is calculation between the values in the Word Representations Matrix and the values contained in the 2x7 Filter Matrix. That generates a new matrix called the Matrix Activation Map, as shown in Fig. 6. Each Matrix Filter has different Activation Maps sizes depending on the size of its Filter Matrix (see Fig. 3(c)). For example, a 2x7 Filter Matrix will produce 1x6 Activation Map Matrix derived from six times sliding from the Filter Matrix.

After the sliding processes are completed, then Matrix Activation Map should be applied normalization by using Rectified Linear Unit (ReLU). In this part will be processed

by converting the minus value to 0. Since there is not a minus value in the sliding results (see Fig. 7(a)), the result of ReLu is the same matrix (see Fig. 7(b)). The result from ReLU will pass the Max-pooling. The goal of the Max-pooling process is finding the largest value, so the largest value is 0.14 (see Fig. 7(c)).

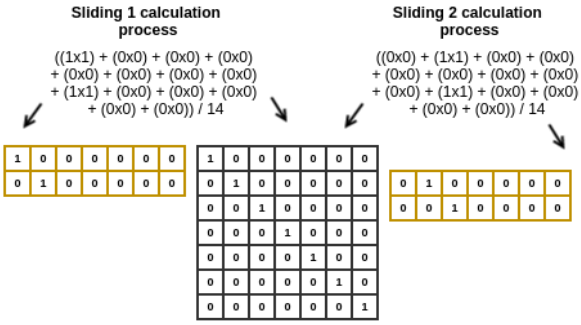

Fig. 5 Sliding process.

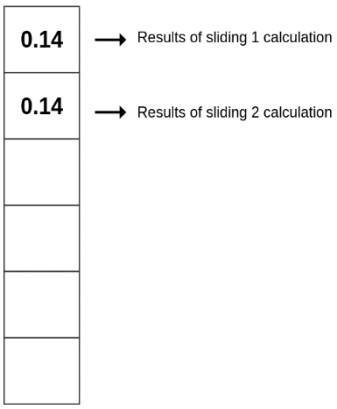

Fig. 6 Results of sliding 1 and 2 processes.

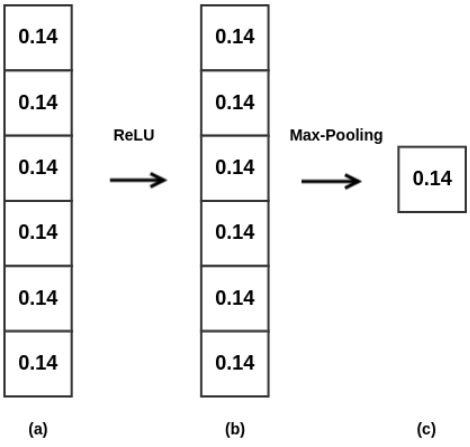

Fig. 7 ReLU and Max-pooling processes.

When the Max-pooling process is passed, then the next process is in the Layer Classification. In this layer, there is Fully-Connected (see Fig. 3(d)), which serves as the place of the Max-pooling process of the Matrix Activation Maps. The number of squares in Fully-Connected appropriate to the total Filter Matrix used in the Feature-extraction Layer. Since the total Filter Matrix used is six, then the number of boxes is six. Each box in Fully-Connected represents every result of the Max-pooling process of the Activation Maps Matrix (see Fig. 8). The value of 0.14 in Fig. 8 is the Max-pooling result of the Matrix Activation Maps generated by the 2x7 Filter Matrix. Max-pooling results from the Matrix Activation Maps generated by other Filter Matrix will be placed in the other Fully-Connected boxes.

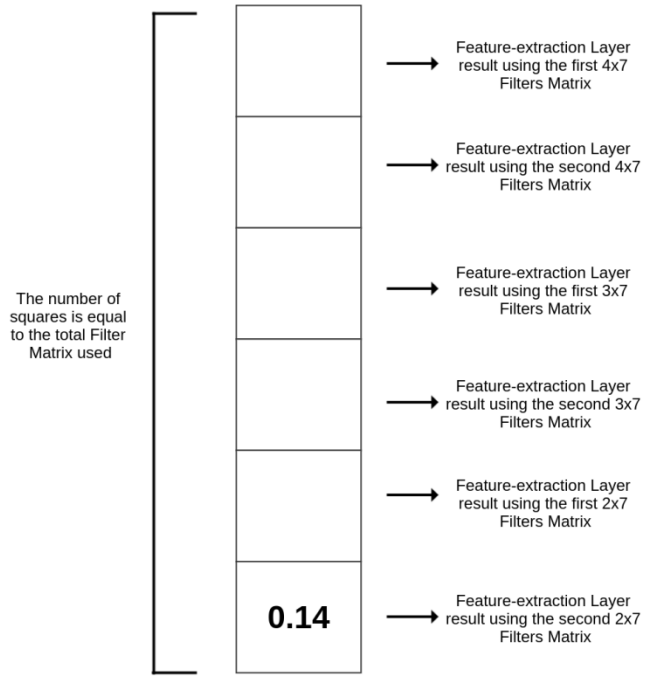

Fig. 8 Value input process in Fully-Connected.

When Fully-Connected is filled in, there is a feed-forward-propagation process from Fully-Connected to the 2x1 matrix (see Fig. 3(d) and Fig. 3(f)**)**. The feed-forward-propagation process will use a randomly-named of weight values (see Fig. 3(e)).

After the feed-forward-propagation process produces the values of the 2x1 matrix, that values will be calculated by the value of Ground Truth (true value) of the sentence "I Like This Movie Very Much!" that are 1 and 0 which mean positive sentiment. The result of the calculation will produce a loss value. If the 2x1 matrix is 0 and 1, it means negative sentiment.

When the loss values have been obtained, it will be continued with running the back-propagation. This back-propagation process will update the weight values of Fully-Connected, 2x1 Matrix (see Fig. 3(e)), and the values in the Filter Matrix (see Fig. 3(b)). After updating the values, the sliding process will be repeated until the calculation process of the loss value and the lowest value is obtained. After getting the lowest loss value, the process and the model are completed. This model will be used in the testing.

III. Data Sets and Experimental Setup

This paper used the Movie Review (MR) and Stanford Sentiment Treebank-2 (SST-2) data sets. The MR data set is a

collection of positive and negative film reviews from the rottentomatoes.com movie review website. Positive and negative reviews have been separated into two different files. The positive review file contains a set of positive movie review sentences and the negative review file contains a set of negative movie review sentences [9].

The SST-2 data set is a positive and negative tweet from https://nlp.stanford.edu. Positive and Negative tweets were also separated into two different files [10]. Examples of data sets can be seen in Table I.

TABLE I
MOVIE REVIEW AND SST-2 DATA SET

	Negative	Positive
MR	simplistic, silly, and tedious	take care of my cat offers a refreshingly different slice of asian cinema
	it's so laddish and juvenile, only teenage boys could possibly find it funny	this is a film well worth seeing, talking and singing heads and all
SST-2	@apple needs to hurry up and release #iTunesMatch	@RIM you made it too easy for me to switch to @Apple iPhone. See ya!
	Why is #Siri always down @apple	I just realized that the reason I got into twitter was ios5 thanks @apple

The pre-training data set above used one hot encoding and word2vec [11]. The frameworks used for training were theano and tensorflow. Theano was developed by the machines learning group of the Universite de Montreal [12], while tensorflow is a framework developed by Google Brain team from Google [13].

When doing the training, this study used cloud technologies that were Google Cloud and FloydHub. Google Cloud is a cloud service provided by Google, to utilize that service, it was needed to use Google's machine learning framework called tensorflow [14] or FloydHub providing various framework such as theano, tensorflow, keras, etc [15].

IV. RESULT AND DISCUSSION

The training processes on theano and tensorflow frameworks were done by using four different parameters settings. Parameters Setting I using embedding dimension: 128 & 300, filter size: 3,4,5, number of filter size: 128, dropout: 0.5, L2 regularization: 0.0, batch size: 64, number of epoch: 200, and data set: movie review (5000 rows of sentence). The results are shown in Table II and Table III,

TABLE II
RESULT OF PARAMETERS SETTING I

Framework	Accuracy	Time
TF	74.1%	2m4s
TF + w2v	79.4%	14m24s
TH+ w2v	80.9%	53m

TABLE III
COMPARE WITH PREVIOUS RESEARCH

Framework	Our Result	Accuracy from previous research			
TF	**74.1%**	*76.1% [4]*	68.1% [7]	*82 .3% [16]*	45.4% [8]
TF+w2v	**79.4%**	*81.0%[4]*	68.1% [7]	*82 .3% [16]*	45.4% [8]
TH+w2v	**80.9%**	*81.0%[4]*	68.1% [7]	*82 .3% [16]*	45.4% [8]

TF: Tensorflow**,** TH: Theano**,** w2v: *Pre-Training* Word2vec

Table III shows the results of the proposed method with the frameworks TF, TF + w2v, TH + w2v (marked with bold numbers) are better, that are, respectively, 74.1%, 79.4%, and 80.9%, than previous studies (indicated by shaded figures), i.e., 68.1% and 45.4% respectively [7], [8]. It occurred since the previous study, it did not use pre-training word2vec and there was no mention of the used framework [7]. The CNN architecture with three Convolution Layers that were Convolution Layer 1, 2, and 3 (see Fig. 9) by adding the word2vec pre-trainer [8]. This is the reason why the results in the proposed study are also better, because of the CNN architecture with three Convolution Layers is too many to handle the data set whose row number is small (5000 lines of the sentence).

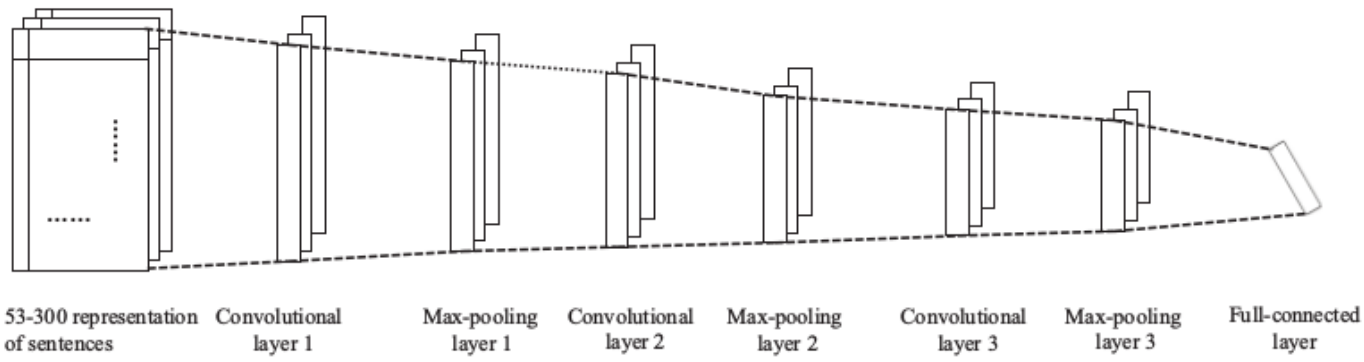

Fig. 9 CNN Architecture from previous study [8].

Table III also shows that the results of the proposed method (marked with bold numbers) have a lower accuracy of 74.1%, 79.4%, and 80.9% compared with results (76.1%, 79.4%, 80.9%) [4] and (82.3%) [16] (indicated by numbers in italics). It occurred because the previous study used the value of L2 regularization that was different from the proposed method [4], while another research not only used the CNN method but also incorporated with the LSTM method [16]. The combined method provided a better performance for sentiment analysis training [16].

The next experiment was using Parameters Setting II with embedding dimension: 300, filter size: 3,4,5, number of filter size: 100, dropout: 0.5, L2 regularization: 0.15, batch size: 64, number of epoch: 25, and data set: movie review (5000 rows of the sentence). The results of the use of Parameters Setting II are shown in Table IV and Table V.

TABLE IV
RESULT OF PARAMETERS SETTING II

Framework	Accuracy	Time
TF	74.8%	8m43s
TF + w2v	73.7%	9m22s
TH+ w2v	80.45%	6m66s

TABLE V
COMPARE WITH PREVIOUS RESEARCH

Framework	Our Result	Accuracy from previous research			
TF	**74.8%**	*76.1%[4]*	68.1% [7]	*82 .3% [16]*	45.4% [8]
TF + w2v	**73.7%**	*81.0%[4]*	68.1% [7]	*82 .3%[16]*	45.4% [8]
TH+ w2v	**80.45%**	*81.0%[4]*	68.1% [7]	*82 .3%[16]*	45.4% [8]

TF: Tensorflow**,** TH: Theano**,** w2v: *Pre-Training* Word2vec

Based on Table V, the proposed study (marked with bold numbers) is still better at 74.8%, 73.7%, and 80.45% than previous studis (indicated by shaded figures) of 68.1% and 45.4% [7], [8]. Table V also shows that the results of the proposed study (marked with the number in bold) of 74.8%, 73.7%, and 80.45% are still lower in accuracy compared to the results of previous research (76.1%, 79.4%, 80.9%) [4] and (82.3%) [16] (indicated by numbers in italics). It shows that changing parameters setting by reducing the number of filter size, adding L2 regularization, and adding the number of epoch do not provide accuracy improvement for the proposed study.

The next experiment was using Parameters Setting III with embedding dimension: 300, filter size: 2,3,4, number of filter: 100, dropout: 0.5, L2 regularization: 0.5, batch size: 50, number of epoch: 25, data set: movie review (5000 rows of sentence). The results of the accuracy are shown in Table VI and Table VII.

If Parameters Setting I and II only made the proposed study provide better accuracy than the previous studies [7], [8] (see Table III and Table V), Parameters Setting III (see Table VII) was capable of producing better accuracy than another study [4], especially using the tensorflow framework of 76.1% (indicated by underlined numbers). It was because the filter size used in Parameters Setting III was smaller (2, 3, 4) than the filter size in Parameters Setting I and II (3, 4, 5). This result is consistent with the previous study, i.e. a smaller filter size has a capability in producing the higher accuracy because it causes the word will be more processed [7].

TABLE VI
RESULT OF PARAMETERS SETTING III

Framework	Accuracy	Time
TF	78.1%	9m45s
TF + w2v	78.9%	8m34s
TH+ w2v	80.25%	6m66s

TABLE VII
COMPARE WITH PREVIOUS RESEARCH

Framework	Our Result	Accuracy from previous research			
TF	**78,1%**	76,1%[4]	77.4%[7]	*82 .3%[16]*	45.4%[8]
TF + w2v	**78,9%**	*81,0%[4]*	77.4%[7]	*82 .3%[16]*	45.4%[8]
TH+ w2v	**80,2%**	*81,0%[4]*	77.4%[7]	*82 .3%[16]*	45.4%[8]

TF: Tensorflow**,** TH: Theano**,** w2v: *Pre-Training* Word2vec

The next experiment used Parameters Setting IV which was applied to the Twitter data set (100,000 rows of the sentence). The Twitter data set has a larger data row than the movie review data set. This Parameters Setting IV used the same parameters setting values as Parameters Setting III since Parameters Setting III has been shown to provide better results than Parameters Setting I and II. The results of the accuracy are shown in Table VIII and Table IX.

TABLE VIII
RESULT OF PARAMETER IV

Framework	Accuracy	Time
TF	79.3%	66m48s
TF + w2v	81.10%	72m57s
TH+ w2v	80.67%	110m9s

TABLE IX
COMPARE WITH PREVIOUS RESEARCH

Framework	Our Result	Accuracy from previous research		
TF	**79.3%**	*82.7% [4]*	*82.3% [17]*	*88 .3% [16]*
TF + w2v	**81.10%**	*86.8% [4]*	*82.3% [17]*	*88 .3% [16]*
TH+ w2v	**80.67%**	*86.8% [4]*	*82.3% [17]*	*88 .3% [16]*

TF: Tensorflow, TH: Theano, w2v: *Pre-Training* Word2vec

Although Parameters Setting IV and Parameters Setting III have the same parameters setting values, they have not able yet to make the proposed study (indicated by bold numbers) getting the greater accuracy (79.3%, 81.10%, 80.67%). The higher accuracies are (82.7%, 86.8%) [4], (82.3%) [17], and (88.3%) [16], indicated by numbers in italics), as shown in

Table IX. It shows that although the Parameters Setting III and IV values were able to produce a better accuracy on the movie review data set, the parameters setting is not capable in producing better accuracy when applied to the Twitter data set. Thus, the parameters settings used were strongly influenced by the characteristics of the data set used, such as the number of data set rows. In addition to parameters setting, accuracy was also influenced by the CNN architecture used, as well in previous studies which used a more complex CNN architecture on the twitter data set and got the better results [16], [17].

In terms of the length of training time, there were differences between movie review data set and Twitter data set. The Twitter data set took longer time in the training process. It was because Twitter data sets had more rows of a sentence than movie review data sets (see Table X).

TABLE X
COMPARE TIME PROCESSING OF DATA SET

Framework	Movie Review (5000 rows)	Twitter (10000 rows)
TF	9m45s	66m48s
TF + w2v	8m34s	72m57s
TH+ w2v	6m66s	110m9s

V. CONCLUSION AND FUTURE WORK

From the results of the proposed experiments, it can be concluded that, firstly, CNN architecture with too many Convolution Layer can decrease accuracy, because CNN architecture was not appropriate for handling data sets with small data rows. Therefore, to determine the CNN architecture, the amount of available data needs to be considered. Secondly, smaller Filter Matrix size provides higher level of accuracy than larger Filter Matrix size. It was because the word will be more processed. Thirdly, the use of word2vec pre-training provides an improved accuracy because it considers the position and context of a word in a sentence. That is different from one hot encoding which is only based on word position in a sentence only.

For the future research, research development using more complex CNN architectures is necessary to handle large data.

REFERENCES

[1] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. New Jersey: Prentice-Hall Inc, 1994.

[2] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul: West Publishing Co., 1992.

[3] D. Puspatiningrum, *Pengantar Jaringan Saraf Tiruan*. Yogyakarta: Penerbit Andi, 2006.

[4] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[5] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *arXiv:1510.03820 [cs.CL]*, 2015.

[6] A. Hassan and A. Mahmood, "Deep Learning Approach for Sentiment Analysis of Short Texts," *2017 3rd Int. Conf. Control. Autom. Robot*, 2017, pp. 705–710.

[7] K. G. Pasi and S. R. Naik, "Effect of Parameter Variations on Accuracy of Convolutional Neural Network," *Int. Conf. Comput. Anal. Secur. Trends, CAST 2016*, 2017, pp. 398–403.

[8] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment Analysis Using Convolutional Neural Network," *2015 IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secur. Comput. Pervasive Intell. Comput.*, 2015, pp. 2359–2364.

[9] B. Pang, (2002) "Movie Review Data," [Online], https://www.cs.cornell.edu/people/pabo/movie-review-data/, accessed: 20-May-2017.

[10] R. Socher, (2013) "Stanford Sentiment Treebank," [Online]. Available: https://nlp.stanford.edu/sentiment/, accessed: 20-May-2017.

[11] (2013) "Word2vec," [Online], https://code.google.com/p/word2vec/, accessed: 21-May-2017.

[12] (2017) "Theano," [Online], http://deeplearning.net/software/theano/index.html, accessed: 22-May-2017.

[13] (2017) "Tensorflow," [Online], https://www.tensorflow.org/, accessed: 22-May-2017..

[14] (2017) "Google Cloud," [Online], https://cloud.google.com, accessed: 21-May-2017.

[15] (2016) "FloydHub," [Online], https://www.floydhub.com, accessed: 22-May-2017.

[16] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, Vol. 72, pp. 221–230, 2017.

[17] J. D. Prusa and T. M. Khoshgoftaar, "Improving Deep Neural Network Design with New Text Data Representations," *J. Big Data*, Vol. 4, No. 7, pp. 1-16, 2017.