Sentiment classification of delta robot trajectory control using word embedding and convolutional neural network

Zendi Iklima, Trie Maya Kadarina, Muhammad Hafidz Ibnu Hajar
Department of Electrical Engineering, Faculty of Engineering, Mercu Buana University, Jakarta, Indonesia

Article Info

Article history:

Received May 4, 2021
Revised Jan 15, 2022
Accepted Feb 4, 2022

Keywords:

Convolutional neural network
Delta robot
Navigation command
Sentiment classification
Word embedding

ABSTRACT

Sentiment classification (SC) is an important research field in natural language processing (NLP) that classifying, extracting and recognizing subjective information from unstructured text, including opinions, evaluations, emotions, and attitudes. Human-robot interaction (HRI) also involves natural language processing, knowledge representation, and reasoning by utilizing deep learning, cognitive science, and robotics. However, sentiment classification for HRI is rarely implemented, especially to navigate a robot using the Indonesian Language which semantically dynamics when written in text. This paper proposes a sentiment classification of Bahasa Indonesia that supports the delta robot to move in particular trajectory directions. Navigation commands of the delta robot were vectorized using a word embedding method containing two-dimensional matrices to propose the classifier pattern such as convolutional neural network (CNN). The result compared the particular architecture of CNN, GloVe-CNN, and Word2Vec-CNN. As a classifier method, CNN models trained, validated, and tested with higher accuracy are 98.97% and executed in less than a minute. The classifier produces four navigation labels: right means *'kanan'*, left means *'kiri*', top means *'atas*', bottom means *'bawah*', and multiplier factor. The classifier result is utilized to transform any navigation commands into direction along with end-effector coordinates.

Corresponding Author:

Zendi Iklima
Department of Electrical Engineering, Faculty of Engineering, Mercu Buana University
South Meruya, Kembangan, Jakarta, Indonesia
Email: zendi.iklima@mercubuana.ac.id

1. INTRODUCTION

A multi-robot communication effectively carries out more tasks instead of robot-to-robot communication. A multi-robot coordination system is feasible to control motion-coordination tasks for large-scale swarm robots [1]. Multi-robot systems may be used in various circumstances, including drone delivery, agricultural, search-and-rescue, disaster relief, and defense [2]. Nowadays, robots are able to communicate not only to other robots but also communicate to humans. Socially interactive robots must be capable of participating in human-like discussions. In this context, human-robot interaction (HRI) also involves natural language processing, knowledge representation, and reasoning [3].

Sentiment classification is an intriguing and essential research issue in natural language processing (NLP) [4]. The sentiment classification task obtains the sentiment polarity contained in the sentence while the sentence information contained in the word is incomplete [5]. The sentence encoder is required to extract features of the sentence to generate the vector representation of the sentence. Sentiment classification attempts to classify opinionated material automatically which consists of analyzing opinions [6], evaluations, emotions, attitudes, evaluations, and other sentiment entities. Although the literature on sentiment classification is fairly rich, sentiment classification for HRI is seldom applied, particularly when navigating a robot using the Indonesian language, which is semantically dynamic when expressed in text.

An Indonesian language sentiment classification has been implemented in a lot of applications such as movie reviews, presidential elections [7], and hotel reviews. Certain approaches utilized as sentiment models such as Naive Bayes (NB), support vector machine (SVM) [8], and other deep learning models such as deep neural network (DNN) [9], and convolutional neural network (CNN) [10]. Word embedding is a technique that learns continuous low-dimensional vector space of word representations by using contextual information from a substantial corpus [11]. The vector representation of words using unsupervised techniques [12] has proven to be very effective in explaining the meaning of the sentiment. Word embedding has several types such as Word2Vec, GloVe [13], Re(Word2Vec), and Re(GloVe) [14]. To enhance the performance of distinguishing words, the word embedding method was improved into deep learning model performance such as CNN, and bidirectional long-short-term memory (Bi-LSTM).

Delta robot trajectory planning has been studied using particular methods such as numerical algorithm, geometrical method, genetic algorithm (GA), fuzzy algorithm, particle swarm optimization (PSO), and artificial neural network (ANN) [15]. However, none of the methods integrate with sentiment classification. Sentiment classification for HRI is rarely implemented, especially for navigating delta robot trajectory planning. The Delta robot has been selected as the research object that manages the trajectory point in the coordinate space of the end-effector (x, y).

Therefore, this paper proposed a sentiment classification of Bahasa Indonesia to produce the Delta robot trajectory planning using word embedding and CNN. The word embedding (Word2Vec and GloVe) produces text patterns as an initial feature of CNN inputs with n text dimension. The CNN classifier produces four navigation labels: right means *'kanan'*, left means *'kiri'*, top means *'atas'*, bottom means *'bawah'*, and multiplier factor. The output of the CNN classifier to support the direction of delta robot trajectory planning. By giving the trajectory set point, the delta robot is able to navigate the subsequent successive trajectory within the text command.

2. RESEARCH METHOD

The self-learning of the delta robot has been developed using Inverse Kinematics (IK) and artificial neural networks (ANN) [15]. This research is to improve the input by providing a sentiment classification of given semantics in Bahasa Indonesia which involves three stages which contain dataset collection, dataset pre-processing, and classification process. The desired output provides the navigation label to the ANN layers. Figure 1 shows the Bahasa Indonesia sentiment classification research method to produce the delta robot trajectory planning using word embedding and CNN.

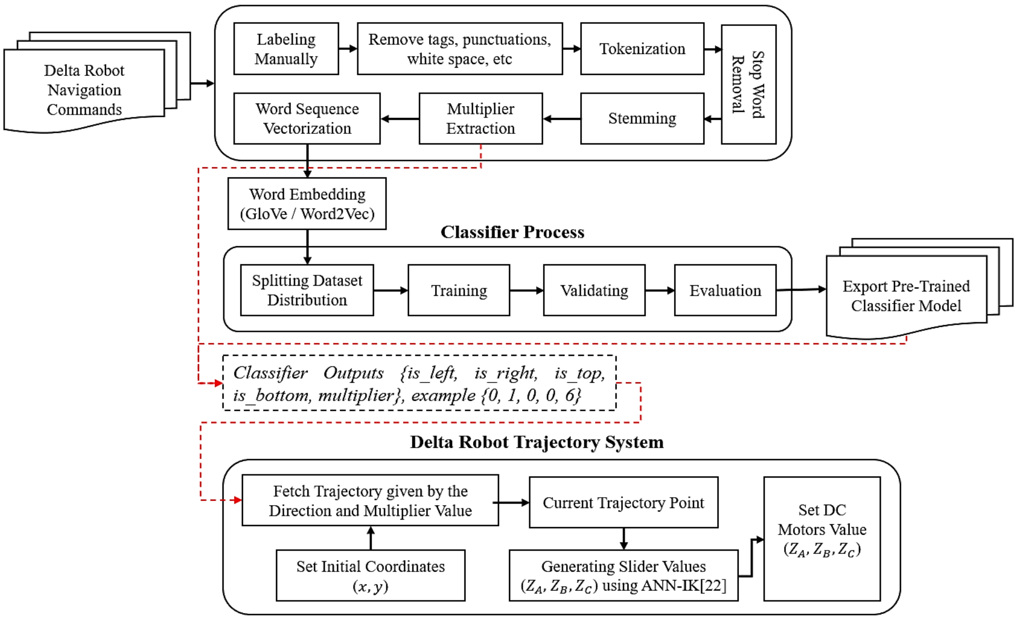


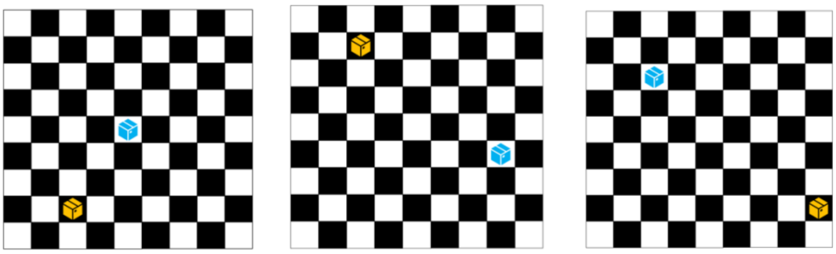
Figure 1. Research method

Our dataset was collected by distributing the questioner to 46 people. The navigation commands are manually labeled into four labels. The Pre-processing stage removes noise inputs to the classifier model to produce high accuracy of the classifier. Furthermore, we embed the clean navigation command to the word embedding vectors such as Glove and Word2Vec. This stage produces 50 dimensions of the image vector. This paper tuning the CNN hyperparameter to analyze the preferred CNN model given by the dataset distribution on the training, validation, and testing processes. Additionally, this paper visualizes the well-tuned CNN models to the Delta robot trajectory control, which utilizes the IK-ANN to navigate the end-effector coordinates given by the sentiment classifier.

2.1. Dataset collection method

Questionnaire distribution proposed to collect delta robot navigation commands dataset. navigation commands collected consist of four navigation labels: left *'kiri'*, right *'kanan'*, up *'atas'*, and down *'bawah'*. Figure 2(a) represents the questionnaire image form for model 1, Figure 2(b) represents the questionnaire image form for model 2, and Figure 2(c) represents the questionnaire image form for model 3. The Quistionnaire image utilized to help the audience filling the navigation command form.

Table 1 shows the navigation commands to Figure 2, these collected commands are utilized to navigate the delta robot to pick up the yellow box to the coordinate of the blue box. The navigation commands collected in 2024 rows of particular command labels containing: 516 rows of left '*kiri*' command, 496 rows of right *'kanan'* command, 598 rows of top '*atas*' command, and 414 rows of bottom '*bawah*' command. The proposed method will be performed using delta robot navigation dataset.



(a) (b) (c)

Figure 2. Questionnaire image of Delta robot navigation commands (a) model 1, (b) model 2, and (c) model 3

2.2. Dataset preprocessing

Table 1 shows the dataset of delta robot navigation commands collected by distributing the questionnaire of given images. The navigation command that was collected was stored as unstructured sentences and unable to classify directly. Unstructured data is hard to classify and yields poor accuracy. Raw sentences require to be preprocessed to produce clean sentences as a classifier input [16]. In Indonesian langeage, a particular sentence can be derived denotative sentence, for instance, a navigation command "*satu kotak kekanan*', '*kekanan satu langkah*', '*pindah kekanan*' that means '*one step to the right*'. The desired sentence refers to '*ke kanan'* checked on Indonesian Dictionary or *Kamus Besar Bahasa Indonesia* (KBBI).

Preprocessing contains sub-process such as manually labeling, case folding, removing punctuations, removing a single character, removing white space, replacing slang words, tokenization, removing stop-word, stemming, and word vectorization [17]. The aim of text preprocessing is to remove a noise input to the classifier method to produce high accuracy containing case folding, remove slang words, remove pructuations, tokenization, stop word removal, stemming, multiplier extraction, and word vectorization.

Case folding converts input sentences into lowercase without exception. Replacing slang words when the audience writes *'kekanan'*, *'kknan'*, *'knn'*, *'kanaaaaaannn'* is wrong, because it should be written as *'ke kanan'*. On the other hand, most of the audience wrote *'sangat'*, *'banget'*, *'bgd'*, *'bangeeedddd'*, and *'...sekali'*, *'bangedh'*, which mean *'really'*. This process replaces slang words into a word that is found in the Indonesian dictionary (KBBI). Furthermore, another process is remove punctuations and white space such as ~!@#$%^&*()_+{}:"?><;'/.,[]\=-`. Tokenization is a process of segregating a sentence into a word vector separated by space. e.g '*one step to the right*' or '*satu langkah kekanan*' into '*satu*', '*langkah*', and '*kekanan*'. Stop word removal is a process of searching and removing certain words that do not affect the desired classifier, such as conjunctions [18] (e.g 'dan' means 'and', 'lalu' means 'then '*selanjutnya*' means '*furthermore*'). Stemming is a process to returns the original form of the word, which contains prefixes and suffixes [19]. For instance, '…*ke kanan*' into '…*kanan*' means '*...to the right*', '…*ke kiri*' into '…*kiri*' means

'*…to the left*', '*…ke atas*' into '*atas*' means '*…upward*'. Multiplier extraction is a process to extract the multiplier command given by the stemming outputs such as '*satu*' into '1' mean '*one*', '*dua*' into '2' mean '*two*'. Word Vectorization is a process of vectorizing/weighting a word into random sequence word, for instance, '*kanan dua langkah*' into (2, 5, 7), and '*empat kanan*' into (6, 2). Word vectorization optimizing the word representation/word embedding method into a dimension of the word.

Table 1. Collected navigation commands

Commands	Labels			
	left 'kiri'	*right* 'kanan'	*top* 'atas'	*bottom* 'bawah'
Delta Robot, please move the yellow box one step to the right • '*Delta robot, pindahkan kotak kuning 1 langkah kekanan*' • '*Delta robot, pindahkan kotak kuning satu langkah ke kanan*' • '*pindahkan ke kanan satu kotak*' • '*diawali dgn memindahkan 1 kotak ke kanan*' • …..	0	1	0	0
Then, two boxes up • *'kemudian, 2 kotak ke atas'* • *'kemudian, 2 kotak keatas'* • *'lalu, 2 kotak ke atas'* • *'2 kotak ke atas'* • ….	0	0	1	0
Furthermore, one step to the right • *'lalu, satu box ke kanan'* • *'dan, satu langkah kekanan'* • *'satu kotak ke kanan'* • *'knan 1'* • …	0	1	0	0
Finally, drop the yellow box by moving one box up • *'terakhir, satu box ke atas'* • *'terakhir, pindahkkan satu kotak ke atas'* • *'kemudian, satu langkah ke atas sampai di box biru'* • *'pindahkan kotak kuning satu langkah ke atas, sampai di kotak biru'* • …	0	0	1	0

2.3. Word embedding and classifier

Convolution is a common matrix multiplication that consists of a convolution layer, pooling layer, and fully connected layer. CNN's inherently handle variable-size sentences through pooling operations, and they also consider the order of words and the context in which each word appears. An opposite understanding of CNN in image classification, where a high number of convolutional layers such as *VGG*, *AlexNet*, *MobileNet*, and *ResNet*, are used. These networks include many neural networks that extract abstract features from images and require huge memory, huge computational requirements during the training process, and require more computationally intensive networks to produce higher accuracy based on the layers used. In the context of text classification, at least a single convolution layer is sufficient to be implied. Figure 3 illustrates the architecture of a convolutional neural network (CNN).

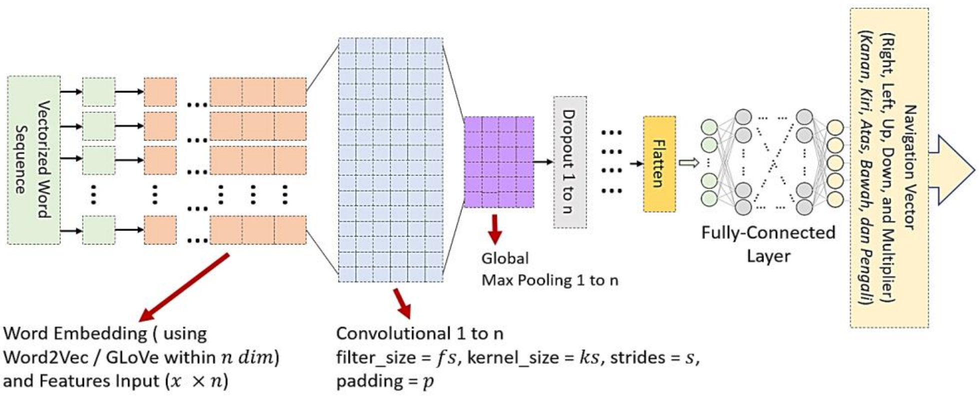


Figure 3. The architecture of convolutional neural network with word embedding

Since $w \in \Re^{d}$denoted as word representation matrices in d-dimensional Word2Vec or GloVe. The convolutional filters denote as$w \in \Re^{hd}$, h denote as window filters, and input matrices denote as $x_{i:i+h-1}$. Therefore, Convolutional filters in a word can be formalized [20].

$$\{x_{1:h}, x_{2:h+1}, \dots, x_{n-h+1:n}\} \tag{1}$$

As shown in (4) produces a feature map $c \in \Re^{n-h+1}$. Moreover, the activation function can be expressed as:

$$c_i = max(\,0, wx_{i:i+h-1}) \tag{2}$$

therefore, the feature map function of the convolution operation can be expressed as (3).

$$c = [c_1, c_2, \dots, c_{i:i+h-1}] \tag{3}$$

Table 2 shows hyperparameters utilized in the performance analysis of our proposed models. Convolutional layers number has assorted hyperparameters such as word embedding dimension, the number of filters, batch size, the number of epochs, and the number of convolutional layers.

Table 2. CNN hyperparameters

Parameters	Values
Word Embedding Dimension	50
Number of Convolution Layers	2, 3, 5, 7, 9
Pooling Layers	GlobalMaxPooling, MaxPooling
Number of Filters	10, 16, 32, 64, 128, 256
Filter Size	2, 3, 5, 7
Number of Fully-Connected Layer [21]	1
Activation Function	ReLu, Softmax
Optimizer	Adam
Dropout [10]	0.6
Regularizer [21]	L2
Batch Size	16, 32, 64, 128, 256
Number of Epoch	50

Performance metrics are utilized to show the capability of the classifier models, such as the accuracy among the dataset distribution for training, validation, and testing. Accuray entails the correct and incorrect predictions of the proposed model, which entails the confusion matrix such as the number true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [22]. The area under the curve (AUC) of the receiver operating characteristic (ROC) or Precision-Recall (PR) curves are quality measures of binary classifiers. Unlike the accuracy, AUC evaluates all the operational points of a model such as sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV) [23].

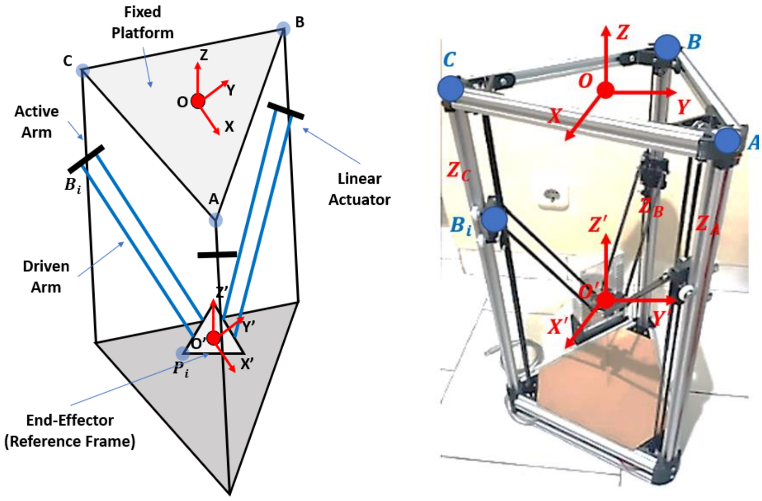
2.4. Delta robot

The delta robot structure commonly constructs fixed and driven frames. $R(O - xyz)$ Illustrates the fixed frame located in the middle of the upper frame ABC. $R'(O - xyz)'$ is a reference frame located in the end-effector frame $P_1P_2P_3$. Z-axis is perpendicular to the end-effector, Y-axis is parallel to P_3O [15]. Figure 4(a) shows the delta robot frame and Figure 4(b) shows the delta robot mechanical design [15], [24].

Inverse kinematics (IK) are used in robot control system to transform join space to angular space [24]. In the delta robot control system, IK inputs related to the end-effector coordinate on the base frame $R'(O - xyz)'$ to achieve direction of the sliders on the Z-axis $Z_AZ_BZ_C$[24] can be formalized as [15],

$$Z_{i,j} = \pm\sqrt{L_i^2 - (x - x_j)^2 - (y - y_j)^2} \tag{4}$$

where $Z_{i,j}$ denotes as the slider direction on Z-axis and $L_i$denotes as the link length between B_iP_i. A self-learning delta robot successfully solved an Inverse Kinematic by using a deep learning approach such as ANN. ANN configures to maintain the prediction of the joint angle of the end-effector given by the coordinates [15].



(a) (b)

Figure 4. Delta robot of (a) frame and (b) mechanical [15], [25]

3. RESULTS AND DISCUSSION

This section shows the result of text preprocessing, text classification results using particular CNN models, and delta robot navigation based on text classification results. The experiment generated 50 dimensions of GloVe and Word2Vec to be trained on proposed models. The CNN models in Table 3, trained with optimal parameters of convolutional layers, filters, and kernels. Meanwhile, selected activation function is 'ReLu', selected regularization is 'Dropout' with value equal to 0.6 [10], and selected batch size 64 [26].

Table 3. Proposed CNN models

Models	Conv Layers	Filters Size	Kernels Size
Model 1 [7]	5	64	3
Model 2 [10]	3	512	6,7,8
Model 3	3	128	5
Model 4 [21]	3	128	7, 4, 3
Model 5 [27]	4	90	2,3,4,5
Model 6 [25]	3	150	3,5,7
Model 7	3	64	3
Model 8	5	128	3
Model 9	7	128	2
Model 10	5	256	2,3,4,5,6

3.1. Text preprocessing results

Figure 5 represents text processing result containing raw text, preprocessing result, and after multiplier extraction. Figure 5(a) shows the raw text containing 516 rows of left '*kiri*' command, 496 rows of right '*kanan'* command, 598 rows of up '*atas*' command, and 414 rows of down '*bawah*' command. Figure 5(b) represent text preprocessing results which consists of the following process such as: case folding, removing punctuation, replacing slang words, tokenization, stop-word removal, stemming, and vectorizing to word embedding matrices. Figure 5(c) represents the word cloud after multiplier extraction, which contains multiplication values such as 1 'one' or 'satu', 2 'two' or 'dua', …., 9 'nine or '*sembilan'*. The multiplier value will be used to multiply the number of steps follows a selected set point in coordinate x and y .

3.2. Classifier performance comparison

The CNN model assessment contains the training and validation process. Therefore, the dataset requires to be partitioned into training datasets and validation datasets (20% of the training dataset). This experiment uses 10 different CNN models. Hence, to compare CNN model performance, particular metrics can be used, such as accuracy, validation of accuracy, AUC, and execution time. Figure 6 shows several CNN observations graphs of accuracy in 50 iterations. Table 4, Table 5, and Table 6 were recorded after training and evaluation for each proposed CNN model ten times.

(a) (b) (c)

Figure 5. Wordcloud of navigation commands for (a) raw text, (b) preprocessing result, and (c) after multiplier extraction

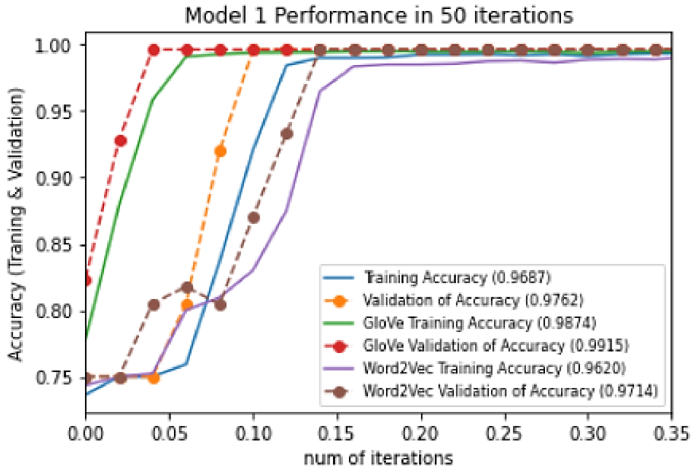


Figure 6. CNN Observation graph of accuracy

Table 4. CNN model performance

Models	Accuracy Tr.	Accuracy Val.	Accuracy Ts.	AUC	Exec. Time (s)
Model 1 [7]	0.9458	0.9532	0.9938	0.9228	17.6681
Model 2 [10]	0.9678	0.9726	0.9938	0.9517	39.7075
Model 3	0.9598	0.9618	0.9938	0.9433	18.4506
Model 4 [21]	0.9732	0. 9772	0. 9938	0.9621	18.3931
Model 5 [27]	0.9671	0. 9698	0. 9938	0.9495	18.2123
Model 6 [25]	0.9705	0. 9701	0. 9938	0.9569	19.5357
Model 7	0.9361	0. 9561	0. 9938	0.9308	16.1871
Model 8	0.9630	0. 9656	0. 9938	0.9393	21.7301
Model 9	0.9676	0. 9717	0. 9938	0.9456	23.1782
Model 10	0.9561	0. 9602	0. 9938	0.9292	48.0551

Table 5. CNN-GloVe model performance

Models	Accuracy Tr.	Accuracy Val.	Accuracy Ts.	AUC	Exec. Time (s)
Model 1 [7]	0.9817	0.9868	0.9876	0.9875	18.8061
Model 2 [10]	0.9897	0.9952	0.9938	0.9952	57.1364
Model 3	0.9851	0.9937	0.9877	0.9989	18.7944
Model 4 [21]	0.9877	0.9925	0.9876	0.9921	19.3583
Model 5 [27]	0.9817	0.9908	0.9938	0.9879	19.2866
Model 6 [25]	0.9872	0.9891	0.9938	0.9929	19.8789
Model 7	0.9667	0.9817	0. 9938	0.9802	17.1706
Model 8	0.9874	0.9917	0.9938	0.9932	22.3027
Model 9	0.9825	0.9779	0.9938	0.9879	23.1683
Model 10	0.9895	0.9925	0.9938	0.9941	68.7105

Table 6. CNN-Word2Vec model performance

Models	Accuracy Tr.	Accuracy Val.	Accuracy Ts.	AUC	Exec. Time (s)
Model 1 [7]	0.9606	0.9718	0.9876	0.9595	18.7656
Model 2 [10]	0.9352	0.9431	0.9938	0.9128	55.7453
Model 3	0.9395	0.9503	0.9876	0.9128	18.4844
Model 4 [21]	0.9674	0.9739	0.9938	0.9673	18.8307
Model 5 [27]	0.9563	0.9673	0.9938	0.9459	18.9545
Model 6 [25]	0.9645	0.9701	0.9938	0.9546	19.1649
Model 7	0.9368	0.9579	0.9937	0.9395	16.9230
Model 8	0.9654	0.9703	0.9937	0.9515	21.3534
Model 9	0.9697	0.9654	0.9939	0.9535	21.7052
Model 10	0.9497	0.9545	0.9936	0.9275	68.9570

Based on Table 4, whole models were observed with batches of the dataset, which contains training dataset (Tr.), validation dataset (Val.), and testing dataset (Ts.). The proposed models perform with great accuracy that is higher than 93% until 97.32%. AUC models are calculated to measure the tradeoff between true-positive and false-positive rates, representing the equality of positive and negative labels. Model 2 and Model 10 were trained longer than the other models because they have a higher filter and kernel size.

Table 4 is the performance comparison between particular CNN models without word embedding matrices. Meanwhile, Tables 5 and 6 show the CNN model embedded by word representation matrices such as GloVe and Word2Vec. By representing preprocessed words into 50 dimensions of word embedding matrices, the weights of standard CNN might be modified by using these matrices. The performance of GloVe-CNN and Word2Vec CNN successfully improves the performance of a standard CNN model.

3.4. System testing

This section explains the whole testing of the classifier CNN and DNN of the delta robot. By selecting three navigation commands based on Figure 1, navigation command sequences pass the text preprocessing step and its classifier. Table 7 shows the extractions of the following results of SA results and its multiplier.

Table 7. Navigation commands testing

No	Seq. Nav. Commands	Clean Txt.	Extract. Nav.	Multiplier
1	move one step to the right ‘*pndah stu ktak kknan*’	‘*kanan satu*’	*‘kanan’* [1, 0, 0, 0]	1
	move two steps to the top ‘*dua ktak keatas*’	‘*atas dua*’	*‘atas’* [0, 0, 1, 0]	2
	to the right ‘*kknan*’	‘*kanan satu*’	*‘kanan’* [1, 0, 0, 0]	1
	to the top ‘*trs keatas*’	‘*atas satu*’	*‘atas’* [0, 0, 1, 0]	1
2	move two boxes to the bottom ‘*gser 2 ktak kbwah*’	*‘bawah dua’*	*‘bawah’* [0, 0, 0, 1]	2
	two boxes to the right ‘*2 ktak knan*’	*‘kanan dua’*	*‘kanan’* [1, 0, 0, 0]	2
	move two boxes to the bottom ‘*gser 2 ktak kbwah lg*’	*‘bawah dua’*	*‘bawah’* [0, 0, 0, 1]	2
	two boxes to the right ‘*2 ktak kknan*’	*‘kanan dua’*	*‘kanan’* [1, 0, 0, 0]	2
3	four boxes to the left ‘*pndah empat step kkri*’	*‘kiri empat’*	*‘kiri’* [0, 0, 0, 1]	4
	move three boxes to the top ‘*pndah 3 step keats*’	*‘atas tiga’*	*‘atas’* [0, 0, 1, 0]	3
	two boxes to the left ‘*duaa step kkri*’	*‘kiri dua’*	*‘kiri’* [0, 0, 0, 1]	2
	two boxes to the top ‘*ats 2 step*’	*‘atas dua’*	*‘atas’* [0, 0, 1, 0]	2

By sequencing sets of navigation commands to move yellow box coordinate into blue box coordinate (shown in Figure 1), the classifier result has given a binary pattern that indicates navigation labels. The direction along the x-axis positive notated as right *'kanan'* direction or in binary representation can be written as 1000. The path along the x-axis negative notated as left *'kiri'* direction or in binary representation can be written as 0100. The path along the y-axis positive notated as top *'atas'* direction or in binary representation can be written as 0010. The direction along the y-axis negative notated as bottom *'bawah'* direction or in binary representation can be written as 0001. The multiplier factor indicates the number of steps that the delta robot moves the set-point coordinate into the end-effector coordinate.

4. CONCLUSION

Human-robot interaction (HRI) successfully integrated using natural language understanding (NLU). Sentiment classification built to classify navigation commands for Delta Robot. Word embedding method (GloVe and Word2Vec) vectorized words into 50-dimensional vectors. These vectors are enabled to be trained in particular CNN Models. CNN Models were evaluated with several layers, filters, and kernels. CNN results contain four labels such as navigation command to the right *'kanan'*, navigation command to the left *'kiri'*, navigation command to the top *'atas'*, and navigation command to the bottom *'bawah'*. Subsequently, the multiplier is extracted for each navigation command input. The multiplier multiplies the step of a given set point in the x-axis and y-axis. CNN models perform great with higher accuracy 98.97% and execution time less than a minute. A transformation from CNN classifier to delta robot z-sliders handled by DNN. DNN input manages set point x and y, multiplier, and CNN labels. Our DNN models also trained with great accuracy equals 96.43%.

ACKNOWLEDGEMENTS

The author expresses gratitude to the Electrical Engineering Department and Research Center of Mercu Buana University, Jakarta, Indonesia. Under the support, the authors have been completed this research.

REFERENCES

[1] M. H. Mohamad Nor, Z. H. Ismail, and M. A. Ahmad, "Broadcast control of multi-robot systems with norm-limited update vector," *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, pp. 1–12, 2020, doi: 10.1177/1729881420945958.

[2] A. Deka, W. Luo, H. Li, M. Lewis, and K. Sycara, "Hiding Leader's identity in leader-follower navigation through multi-agent reinforcement learning," *arXiv:2103.06359v2*, pp. 4769–4776, 2021, doi: 10.1109/iros51168.2021.9636314.

[3] M. Atzeni and D. Reforgiato Recupero, *Deep learning and sentiment analysis for human-robot interaction*, vol. 11155 LNCS. Springer International Publishing, 2018.

[4] R. Afiyati, E. Winarko, and A. Cherid, "Recognizing the sarcastic statement on WhatsApp Group with Indonesian language text," *2017 International Conference on Broadband Communication, Wireless Sensors and Powering, BCWSP 2017*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/BCWSP.2017.8272579.

[5] S. Zhang, X. Xu, Y. Pang, and J. Han, "Multi-layer attention based CNN for target-dependent sentiment classification," *Neural Processing Letters*, vol. 51, no. 3, pp. 2089–2103, 2020, doi: 10.1007/s11063-019-10017-9.

[6] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, 2016, pp. 225–230, doi: 10.18653/v1/p16-2037.

[7] H. Tsaniya, R. Rosadi, and A. S. Abdullah, "Sentiment analysis towards Jokowi`s government using twitter data with convolutional neural network method," *Journal of Physics: Conference Series*, vol. 1722, p. 012017, 2021, doi: 10.1088/1742-6596/1722/1/012017.

[8] J. Samuel, G. G. M. N. Ali, M. M. Rahman, E. Esawi, and Y. Samuel, "COVID-19 public sentiment insights and machine learning for tweets classification," *Information (Switzerland)*, vol. 11, no. 6, pp. 1–22, 2020, doi: 10.3390/info11060314.

[9] D. Apriliani, T. Abidin, E. Sutanta, A. Hamzah, and O. Somantri, "Sentiment analysis for Indonesia hotel services review using optimized neural network," *Journal of Physics: Conference Series*, vol. 1538, no. 1, 2020, doi: 10.1088/1742-6596/1538/1/012060.

[10] O. S. Jaya, H. Murfi, and S. Nurrohmah, "Analysis of convolution neural network for transfer learning of sentiment analysis in Indonesian tweets," *ACM International Conference Proceeding Series*, pp. 18–22, 2018, doi: 10.1145/3239283.3239299.

[11] O. H. and J. Šedo and Additional, "Forward and inverse kinematics using pseudoinverse and transposition method for robotic arm DOBOT," *Intechopen*, vol. 4, pp. 75–94, 2017.

[12] Y. Sharma, G. Agrawal, P. Jain, and T. Kumar, "Vector representation of words for sentiment analysis using GloVe," *ICCT 2017 - International Conference on Intelligent Communication and Computational Techniques*, vol. 2018-Janua, pp. 279–284, 2018, doi: 10.1109/INTELCCT.2017.8324059.

[13] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, "Improving the accuracy of pre-trained word embeddings for sentiment analysis," *arXiv*, Nov. 2017.

[14] L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings for sentiment analysis," *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017, pp. 534–539, doi: 10.18653/v1/d17-1056.

[15] Z. Iklima, M. I. Muthahhar, A. Khan, and A. Zody, "Self-learning of delta robot using inverse kinematics and artificial neural network," vol. 25, no. 3, pp. 237–244, 2021.

[16] A. B. Prasetijo, R. R. Isnanto, D. Eridani, Y. A. A. Soetrisno, M. Arfan, and A. Sofwan, "Hoax detection system on Indonesian news sites based on text classification using SVM and SGD," *Proceedings - 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2017*, vol. 2018-Janua, 2017, pp. 45–49, doi: 10.1109/ICITACEE.2017.8257673.

[17] W. C. F. Mariel, S. Mariyah, and S. Pramana, "Sentiment analysis: A comparison of deep learning neural network algorithm with SVM and Naïve Bayes for Indonesian text," *Journal of Physics: Conference Series*, vol. 971, no. 1, 2018, doi: 10.1088/1742-6596/971/1/012049.

[18] H. A. Santoso, E. H. Rachmawanto, A. Nugraha, A. A. Nugroho, D. R. I. M. Setiadi, and R. S. Basuki, "Hoax classification and sentiment analysis of Indonesian news using Naive Bayes optimization," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, pp. 799–806, 2020, doi: 10.12928/TELKOMNIKA.V18I2.14744.

[19] E. Miranda, M. Aryuni, R. Hariyanto, and E. S. Surya, "Sentiment analysis using sentiwordnet and machine learning approach (Indonesia general election opinion from the twitter content)," *Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019*, vol. 1, no. August, pp. 62–67, 2019, doi: 10.1109/ICIMTech.2019.8843734.

[20] D. Stojanovski, G. Strezoski, G. Madjarov, I. Dimitrovski, and I. Chorbev, "Deep neural network architecture for sentiment analysis and emotion identification of Twitter messages," *Multimedia Tools and Applications*, vol. 77, no. 24, pp. 32213–32242, 2018, doi: 10.1007/s11042-018-6168-1.

[21] S. Rani and P. Kumar, "Deep learning based sentiment analysis using convolution neural network," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3305–3314, 2019, doi: 10.1007/s13369-018-3500-z.

[22] A. A. H. Francis Jesmar P. Montalbo, "Classification of fish species with augmeted data using deep convolutional neural network," *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, 2019, pp. 396–401.

[23] A. A. Ardakani, A. R. Kanafi, U. R. Acharya, N. Khadem, and A. Mohammadi, "Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks," *Computers in Biology and Medicine*, vol. 121, no. March, p. 103795, Jun. 2020, doi: 10.1016/j.compbiomed.2020.103795.

[24] L. Qingsheng and J. Andika, "Analysis of kinematic for legs of a hexapod using denavit-hartenberg convention," *Sinergi*, vol. 22, no. 2, p. 69, 2018, doi: 10.22441/sinergi.2018.2.001.

[25] H. S. Alatawi, A. M. Alhothali, and K. M. Moria, "Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT," *arXiv*, pp. 1–32, 2020.

[26] H. Imaduddin, Widyawan, and S. Fauziati, "Word embedding comparison for Indonesian language sentiment analysis," *Proceeding - 2019 International Conference of Artificial Intelligence and Information Technology, ICAIIT 2019*, 2019, pp. 426–430, doi: 10.1109/ICAIIT.2019.8834536.

[27] B. Zaman, A. Justitia, K. N. Sani, and E. Purwanti, "An Indonesian hoax news detection system using reader feedback and naïve bayes algorithm," *Cybernetics and Information Technologies*, vol. 20, no. 1, pp. 82–94, 2020, doi: 10.2478/cait-2020-0006.

BIOGRAPHIES OF AUTHORS

Zendi Iklima received the M.Sc. degree in Software Engineering from Beijing Institute of Technology in 2018. His research interests include robotics, deep learning, and cloud computing. He has an experience as the Chief Technology Officer of Diaspora Connect Indonesia. Indonesian Diaspora Connect is one of the controversial start-up companies in 2018. The idea of connecting Indonesia is to help the Indonesian government find out about citizens who are studying or living abroad which integrates with Artificial Intelligence. Based on the idea, he achieved 1st place in ALE Hackathon Competition in 2018 which held by Alcatel-Lucent (Jakarta). He can be contacted at email: zendi.iklima@mercubuana.ac.id.

Trie Maya Kadarina received her bachelor's degree in Electrical Engineering from Institut Teknologi Nasional in 2001. She received her master's degree in Biomedical Engineering from the Department of Electrical Engineering, Institut Teknologi Bandung, in 2005. Currently, she is a lecturer in the Department of Electrical Engineering at Universitas Mercu Buana. Her research interests include biomedical instrumentation, electronics and control systems, machine learning, and Internet of Things. She can be contacted at email: trie.maya@mercubuana.ac.id.

Muhammad Hafizd Ibnu Hajar received the B.Eng. Degree in electrical engineering from Universitas Mercu Buana Jakarta, in 2014, the M.Sc. degree in electronics science and technology from Beijing Institute of Technology, China, in 2017. He subsequently joined Universitas Mercu Buana, where he is currently a lecturer in the Department of Electrical Engineering. Hafizd's principal research interest is robotics, electronics and control systems. His current research interests include internet of things and path planning of robots. He teaches some courses such as research methods, telecommunications electronics, robotics and transmission line. In addition, he was also active in Community Service activities and as a journal managing editor of electrical engineering at Universitas Mercu Buana Jakarta. He can be contacted at email: muhammadhafizd@mercubuana.ac.id.