

Object Texture Recognition Based on Grasping Force Data Using Feedforward Neural Network

A. B. Roslan and R. L. A. Shauri

Abstract—A study on a three-fingered robot hand with a 6-axis force/torque sensor and position-based impedance control was developed to execute texture recognition during grasping tasks. Force sensor data from grasping experiments by the robot hand for a bottle and a ball were used as inputs to the recognition algorithm. Moreover, the stiffness coefficient of the impedance parameter was varied to observe the difference of the force data for the different object textures. Based on the analysis results, the input and output of the artificial neural network (ANN), two layers feed forward network for the recognition process have been determined. The ANN simulations were divided into two simulations, first on the different amount of data used in the training and second, the simulation on selecting the suitable training method. Three training methods were chosen for the simulation i.e. Scaled Conjugate Gradient Backpropagation (SCG), Levenberg-Marquardt Backpropagation (LM), and Bayesian regularization Backpropagation (BR). From the experiments, SCG showed significant results with 72.7% accuracy compared to the LM and BR with 71.3% and 68.7%, respectively.

Index Terms—Artificial Neural Network, Impedance control, Object Grasping, Texture recognition, Training method.

I. INTRODUCTION

ROBOTS play a big role in this modern world where a lot of big companies used robotic approach either to solve a simple task or complex task. This encourages more researches on making these robots able to gather multiple sensor measurements and use them for manipulation control.

Force control is essential for many robot manipulators as their practical manipulation tasks are usually associated with the interaction between the end effector of the robot and the environment. In the current trends, robot manipulators are supposed to be more autonomous in which simple motion control is inadequate to obtain a successful execution for a variety of manufacturing tasks where object sizes and positions may vary. Therefore, the control of the physical interaction between the robot end effector and its environment is necessary

for the successful execution of manipulation tasks. A force/torque sensor can be introduced to provide the force measurements for the force control algorithm.

In current trends, there are many researchers that expanded their research on force control with Artificial intelligence (AI) algorithms. Artificial Neural Network (ANN) is a method that has various approaches to approximate nonlinear functions such as Recurrent Neural Network (RNN), Convolution Neural Network (CNN), and Radial Basis Function Neural Network (RBFNN). The basic structure of ANN contains the input layer, output layer, hidden layer, neuron/node, weight, bias, activation function, and learning function. Every type of ANN is different based on the arrangement and selection of the basic structure to suit the requirement of control for a system. Besides, the training method also plays a role in the accuracy of the ANN where it depends on the size of data training and also the speed of the training time. A study by [1] proposed a deep learning method for a robot manipulator to predict object grasping tasks based on monocular depth image. Grasp Prediction Network (GPNs) was used to predict the candidate groups of grasp points while the Grasp Evaluation Network (GENs) was used to evaluate the candidate groups' grasping quality. GPNs were designed based on Convolutional Neural Network (CNN) and Gaussian Mixture Model (GMM). Rani and Kumar [2] developed a neural network-based hybrid force/position method to solve the uncertainties and external disturbances for constraint rigid robot manipulators. They introduced a Radial Basis Function Neural Network (RBFNN) and adaptive bound part to the control model. In a simulation on two degree of freedom (DOF) robot manipulator with position/force tracking control, an adaptive Jacobian and Radial Basis Function Neural Network (RBFNN) was introduced by [3] as a method to achieve accurate force control performance. Reference [4] implemented a study of a three DOF robot manipulator under two constrained conditions which requires the robot control to adapt with the unknown state and calculate the uncertain dynamic online. The methods proposed were tan-type Barrier Lyapunov Function (BLF) and fuzzy NN based on impedance learning, respectively.

Studies on object and texture recognition have also been implemented to improve robot's manipulation task by introducing AI algorithms into their control systems. A study by [5] introduced a new algorithm for texture recognition that included genetic optimization as the optimizer and support vector machine (SVM) which is less sensitive towards velocity and contact force of the tactile sensor. The short linear slide movement (SLSM) and circular exploration movement (CEM)

This manuscript is submitted on 9th August 2021 and accepted on 7th February 2022.

A. B. Roslan and R. L. A. Shauri are with the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor (e-mail: ruhizan@uitm.edu.my)

*Corresponding author
Email address: ruhizan@uitm.edu.my

1985-5389/© 2022 The Authors. Published by UiTM Press. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

were used to collect the velocity and pressure data. Research by [6] proposed a method that combines force control with Reinforcement learning (RL) to learn the contact-rich tasks on position-controlled robots. A six degree of freedom (DOF) manipulator used six-axis force/torque (F/T) sensor with a Mel-frequency Cepstrum Coefficient (MFCC) and time-delay neural network to recognize the clicking sound produced when closing a pen cap [7]. This was done for different sizes of pens to extract the time-series feature.

Besides, a study by [8] proposed an object recognition method using haptic information that was classified using a Gaussian-Bayesian classifier for eighteen types of objects of different materials and shapes. Research by Kaboli and Cheng [9] proposed a new tactile sensor that can be implemented on robotic hand or humanoid robot which can differentiate the material or texture of object through various types of exploratory movement. The development of the object and texture recognition allows the researchers to explore deeper into the application of ANN in robotic system.

In this paper, the relationship between the stiffness coefficient of the impedance parameter and the texture of the objects based on measured force data is discussed. Then, a 2-layer feedforward network was developed using three different training methods which are Scaled Conjugate Gradient Backpropagation (SCG), Levenberg-Marquardt Backpropagation (LM), and Bayesian regularization Backpropagation (BR). The performance of the different training methods in terms of accuracy were compared and discussed. These methods were used to minimize the output error while increasing the accuracy of the model [10]–[12].

II. ROBOT HAND SYSTEM AND IMPEDANCE CONTROL

The development of the robotic hand was implemented in the previous works, which starts from the 3D design of each component to the position control of the hand [13–15]. The 7-DOF robot hand consists of three fingers (F1, F2 and F3) with all joints actuated by DC micro motors equipped with encoders and a new 6 axis force/torque (ATI NANO17) sensor, as shown in Fig. 1. The total DOFs come from the two joints of each finger, including one DOF at the palm.

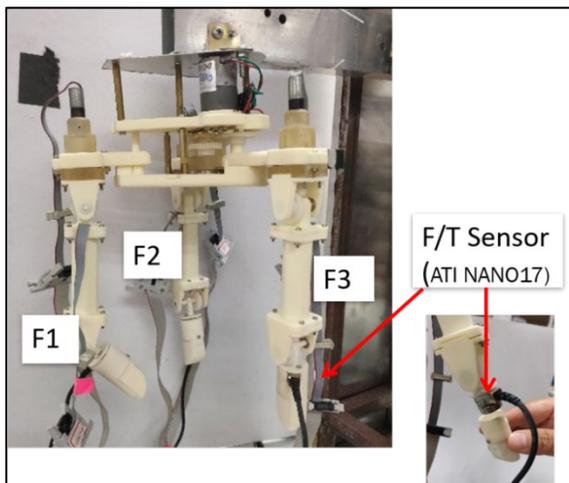


Fig. 1. The three-fingered robot hand

The same kinematics and impedance control equation that was developed by [13] were used, where the dynamic equation for impedance control is as in (1).

$$F_{ext} - F_{ref} = M_d (\Delta \ddot{P}) + D_d (\Delta \dot{P}) + K_d (\Delta P) \quad (1)$$

where, $\Delta P = P_{d_{new}} - P_d$.

Here, the difference between the measured external force, F_{ext} and the reference force, F_{ref} indicates the contact force measured. M_d , D_d , and K_d are the impedance parameters known as mass, damping, and stiffness coefficients, respectively while ΔP denotes the change of the fingertip's position. The impedance control program of the robot hand was verified through simulation and real-time experiments by [14]. The system is highly nonlinear due to the backlash and friction between the motor gears and connection between the links. Adjustments have been made to the hardware but minor effects have been observed to influence the movements of the robot finger during the collection of the force data. Therefore, data filtering has been done to remove the unusable data from being fed to the ANN algorithm.

III. OBJECT TEXTURE RECOGNITION

This work consists of three phases which are experimental and data collection from grasping tasks, analysis on texture recognition, and ANN training. The robot hand was used to execute the grasping task on two different object textures. The execution process of grasping task was done by setting the desired position, P_d for joint 1 and joint 2 where the finger needs to make contact with the test objects. The 6-axis force/torque sensor measured the external force that was exerted to the robot fingers and the rate of force F_{rate} was calculated between the time intervals of 1 second which will be explained in detail in the next subsection. Only the impedance stiffness parameter, K_d was selected and varied for this experiment. The results from the experiment were graphed and analyzed to find the relationship between the impedance parameter and object texture. The ANN simulations were divided into two simulations, i.e. the simulation on different amounts of data used in the training and the simulation on three types of training methods.

A. Phase 1: Run Grasping Tests on Two Objects

In this experiment, joint 1 and joint 2 were set at 10° and 33° degrees, respectively. This allows the robot hand to firmly grasp the selected objects which are the ball and bottle as shown in Fig. 2. These objects were selected based on their hardness texture where the plastic bottle is the hard object while the sponge material ball is the soft object. To differentiate these two types of texture, the F_{rate} as written in (2) was calculated where it measures the difference between the initial force rate F_{ext_i} and the final force rate F_{ext_o} divided by the time taken for the robot hand to grasp the objects. F_{ext_i} is the F_{rate} after 0.25 seconds when the robot hand interacts with the objects while F_{ext_o} is the F_{rate} after 1 second of the grasp.

$$F_{rate} = \frac{F_{ext(1s)} - F_{ext(0.25s)}}{Time} \quad (2)$$

All F_{rate} , F_{ext_i} and F_{ext_o} data were collected from this experiment. As for F_{ext} , it only measures the force in the x-axis

direction of finger 1 which is the direction of the applied external force.

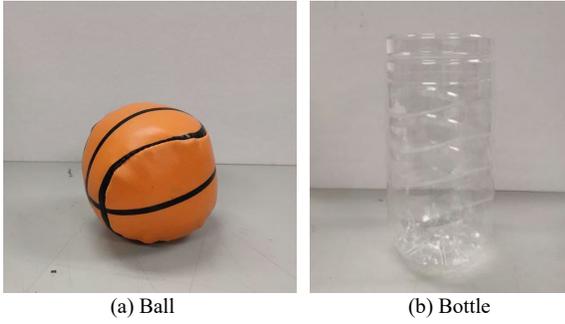


Fig. 2. Selected objects for grasping task

Fig. 3 shows the experiment of grasping a task that was executed on the selected objects. The robot finger moves towards the desired position where the object is grasped and the F_{ext} was measured. All fingers move simultaneously according to the control signals received from the position based impedance control algorithm. In this work, it is the first time for the robot hand to execute the grasping task using the existing control algorithm.

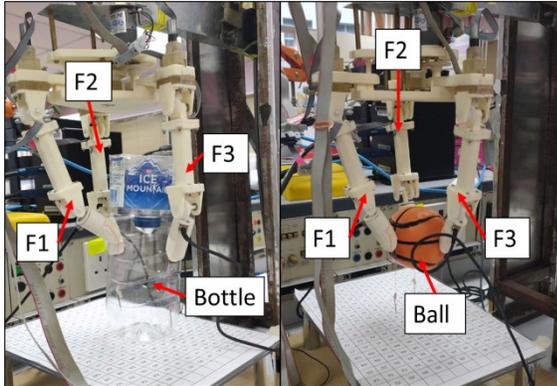


Fig. 3. Setup for grasping tasks

There were two amounts of data used in the simulations where initially 109 data were collected and later an additional of 41 data were collected to increase the number of the experimental data. The K_d for each amount of data was varied between 1000, 500 and 250. The K_d value was chosen between 1000 and 250 because according to Nisa [15] the optimal value of K_d from the previous work on the robot hand using a load cell sensor was 1000. The same K_d value was used to observe the applicability of the new force sensor for the same algorithm.

Fig. 4 shows the illustration process of the texture recognition data collection where it is divided into two steps, the reset and grasping. In the reset step, the initial force for the robot hand was reset to ensure a minimum force error measurement. Next, in the grasping step, the grasping task was executed where the robot hand moves towards the desired position and interacts with the objects. The data was measured when the F_{ext} exceeds the reference force, F_{ref} in the x-axis direction and the F_{rate} was calculated when the grasping exceeds one second.

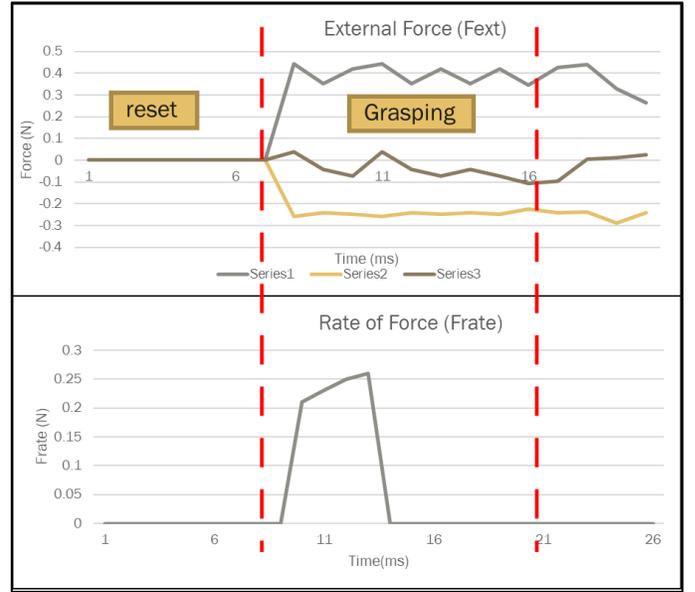


Fig. 4. An illustration process of data collection.

B. Phase 2: Analysis on Texture Recognition

In this section, the analysis on texture recognition is discussed. It consists of the selection of data for analysis, the methods used, and how the data was analyzed. Here, only the F_{rate} data that was used because it was assumed that for an object with stiff or hard texture, the F_{rate} has higher value compared to softer objects. In this case, the bottle was assumed to have a higher value of F_{rate} compared to the ball.

The F_{rate} data were analyzed by using a statistical method where the minimum, maximum, and average of data were calculated. The same average data were used to determine the range of F_{rate} for each K_d . Then, the F_{rate} data for each K_d were plotted using line graph to visualize the relationship between F_{rate} and K_d for different textures. Line graph was used to visualize the data due to its suitability for the data representation.

These data were to be used for the ANN training and testing. The confusion matrix results were used to analyze the accuracy of the training methods.

C. Phase 3: ANN Training

This section discusses the structure, the inputs and outputs, and the simulation setups for ANN. MATLAB pattern recognition tools (patternnet) was used where the default of ANN structure which consists of a two-layer feed-forward network as shown in Fig. 5 is applied. It contains weight w , biases b , activation function, neurons, and layers. Four numbers of layers and two activation functions were used where the layers consist of one input layer, two hidden layers, and one output layer. In addition, this ANN used the supervised learning method where the outputs were trained to recognize the characteristic of the input data. This tool allows the ANN structure to be changed according to the desired performance accuracy of the training. For example, if the desired accuracy for this system is around 90%, it can be achieved by adjusting the number of neurons in the hidden layer, the training method, and the size of data input.

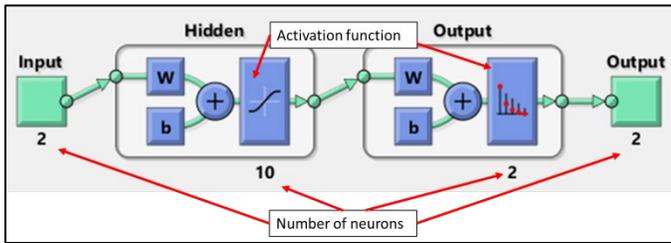


Fig. 5. ANN structure

Table 1 shows the input and output of the ANN, where the inputs are F_{rate} and F_{ext_o} while the outputs are the texture of the objects which is either bottle or ball. The F_{ext_o} data is used in order to increase the input for the ANN.

TABLE I
ANN DATA INPUTS AND OUTPUTS

Input	Output
F_{rate}	Bottle
F_{ext}	Ball

Before training the ANN, the measured data need to be preprocessed by filtering and labeling steps. F_{rate} data were manually filtered by choosing the data accordingly where the faulty data was removed. The faulty data are the data that were obtained when error occurred due to the hardware failure such as robot joint gear and an unexpected finger movement during the collection of data. There were two simulations which have been implemented, firstly was to compare two amounts of data usage for the ANN training to observe the sufficient amount needed to achieve a better accuracy percentage and secondly three types of ANN training methods were used to observe the efficiency of these methods with the given inputs. Table 2 shows the detailed settings that were used in the first and the second simulations. For the first simulation, the number of neurons in the hidden layer varied between 5 to 20 neurons using the same SCG training method. Meanwhile, the second simulation uses the same data but different training methods which are LM and BR backpropagation. There were two types of activation functions used i.e. softmax and sigmoid function, where softmax is used on the hidden layer while sigmoid on the output layer. The sigmoid function was used on the output layer to obtain the range of output value between 0 and 1.

TABLE II
SIMULATION SETTINGS

Details	Simulation 1	Simulation 2
Amount of data use	109 & 150	150
Number of hidden neurons	5,10,15,20	5,10
Training methods	SCG	SCG, LM, BR
Activation Function	Softmax and sigmoid function	

The data distribution was divided into three parts, training, validation, and testing where the percentages of distributions were 70%, 15%, and 15%, respectively. The training was repeated until it could achieve 80% or higher total percentage for the ANN model and it can be concluded that the model was suitable to be used for the real-time experiment. This setup was

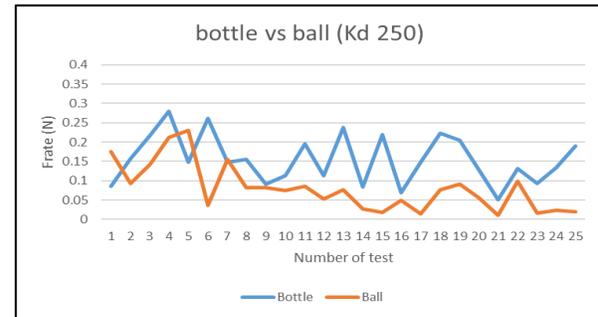
chosen in the MATLAB tools configuration setting and the selection was limited to the list that was provided in the tools.

IV. RESULTS AND DISCUSSION

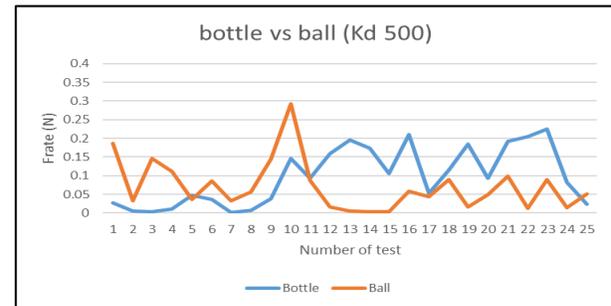
In this section, the first part discusses the data obtained from the grasping experiment, and the second part discusses the results of the ANN simulation.

A. An Experiment of Grasping Task

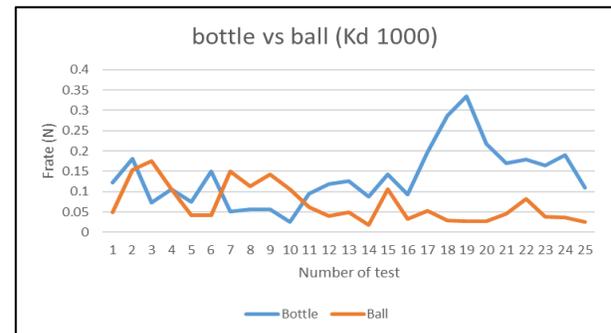
The results from the grasping tasks experiment was obtained and analyzed to confirm the assumption that was made previously. Fig. 6 shows a comparison graph between bottle and ball for K_d 250, 500 and 1000 where the blue and orange lines represent the F_{rate} of the bottle and ball, respectively.



(a): Comparison graph for K_d 250



(b): Comparison graph for K_d 500



(c): Comparison graph for K_d 1000

Fig. 6: Comparison graph for each K_d

From Fig. 6(a), the F_{rate} value for the bottle exceeds the ball for 22 times from the overall 25 times of grasping experiments. Meanwhile, from Fig. 6(b), F_{rate} value of the bottle exceeds the ball by 14 times. Similarly, for K_d 1000, F_{rate} bottle exceeds the F_{rate} ball by 19 times as can be observed in Fig. 6(c). The

graphs show that the F_{rate} bottle for each K_d is higher than the F_{rate} ball.

Table 3 shows the summary of the F_{rate} comparison. This can be concluded that the assumption on higher bottle F_{rate} compared to the ball F_{rate} is true. Besides, the suitable value of K_d is 250, where it allows the robot finger to differentiate between the bottle and ball better based on the value of the F_{rate} . The effects of varying the K_d value will allow the robot's fingertips to apply sufficient force to the objects during grasping.

TABLE III
F_{rate} BOTTLE AND F_{rate} BALL COMPARISON

Value of K_d	F_{rate} comparison		Total compare data
	Bottle > Ball	Ball > Bottle	
1000	19	3	25
500	14	11	25
250	22	6	25

B. ANN simulation results

Fig. 7 shows an example of confusion matrix as the result of ANN simulation which is normally used to describe accuracy performance. The output class corresponds to the rows while the target class corresponds to the columns. There are two colored boxes in each row for this case which indicate the correctly and incorrectly matching results. The diagonal green-colored boxes indicate the correctly matching output class and target class, while the red-colored boxes indicate the incorrectly matching. The most-right column boxes show the percentages of correctly and incorrectly predicted output class to the target class. This term often referred to as positive predictive value and false discovery rate. Meanwhile, the bottom row represents the percentage of the correctly or incorrectly target class to the predicted output class. This term often is referred to as true positive rate and false negative rate.

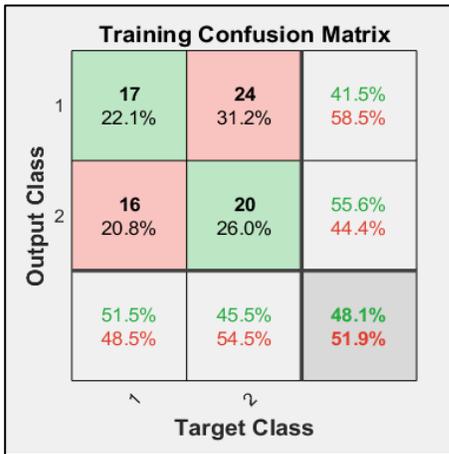


Fig. 7: Example of a confusion matrix

Firstly, the simulation results on 109 data will be presented. Fig. 8 shows the confusion matrix for 10 neurons. The confusion matrices consist of the number of confusions for three states which are Training, Validation, and Test states. The state All is calculated by summing the percentage from all states.

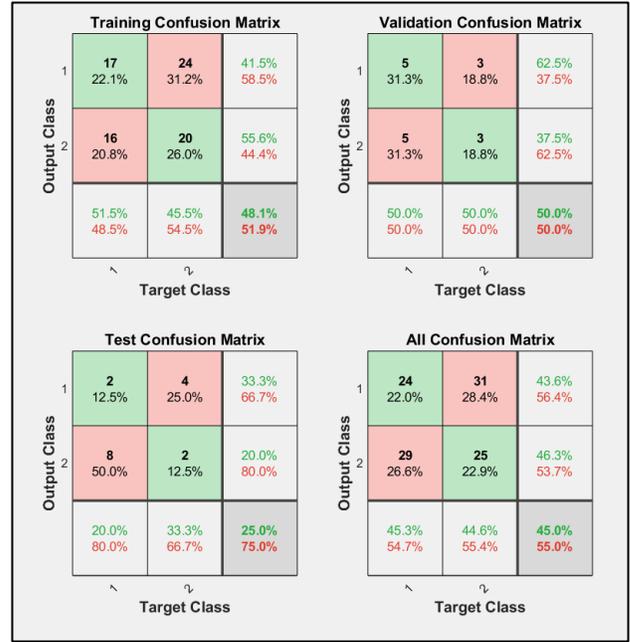


Fig. 8: Example of the confusion matrix for 10 neurons

Fig. 9 shows the comparison between the matching percentage and the number of neurons. Here, the matching percentage refers to the correct matching output class and target class. As indicated in grey line matching percentage, it can be observed that by using 5 neurons, the ANN was able to achieve around 78% which is the highest compared to others for the Test state. However, the All state for 10 neurons was the highest compared to the others. Thus, shows that 10 is the suitable number of neurons to be used for the ANN simulation.

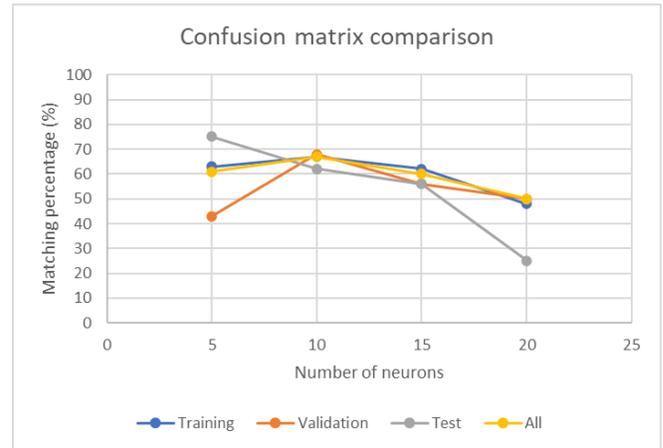


Fig. 9: Comparison number of neurons and matching percentage

For the next simulation, the number of data was increased by 41 additional data which makes a total of 150 data, and each of K_d has the same amount of data, 50.

Fig. 10 shows the confusion matrix for 150 data with 10 neurons where for the Training state, the ANN achieves 74% of correct matching percentage meanwhile the All state percentage drops to 70.7%. The accuracy of the ANN model is observed at All state which means that if the percentage is higher therefore the accuracy can be concluded as higher or otherwise lower.



Fig. 10: Confusion Matrix for 150 data with 10 neurons

Fig. 11 shows the overall performance for 150 data. It can be observed that the 10 neurons result has the highest 70.7% matching percentage for All state compared to other number of neurons.

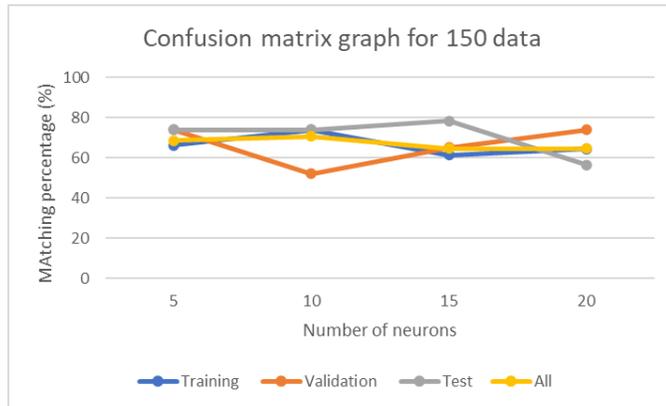


Fig. 11: Comparison of the number of neurons and matching percentage for 150 data.

The comparison between 150 and 109 data with 10 neurons can be observed in Fig. 12. 10 neurons were selected for this due to the accuracy performance from the previous simulation. There was a significant increase in the matching percentage for 150 data compared to 109 data where in the overall performance or All state the difference between two data was 3.7%.

Table 4 shows the result of the matching percentage for each neuron with different amount of data. This simulation shows that if the amount of data could be increased to be more than 150 in the future, higher percentage of accuracy can be potentially achieved.

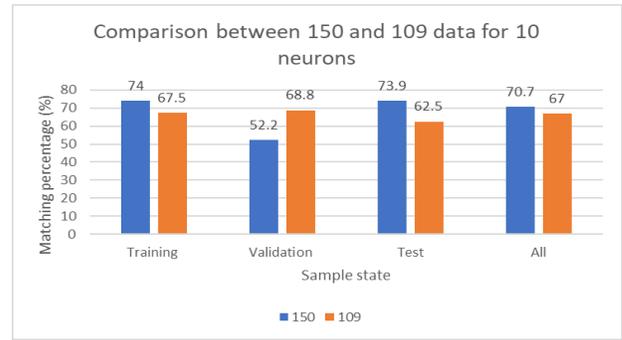


Fig. 12: Comparison graph for 10 neurons

TABLE IV
SUMMARY OF MATCHING PERCENTAGE WITH NUMBER OF DATA

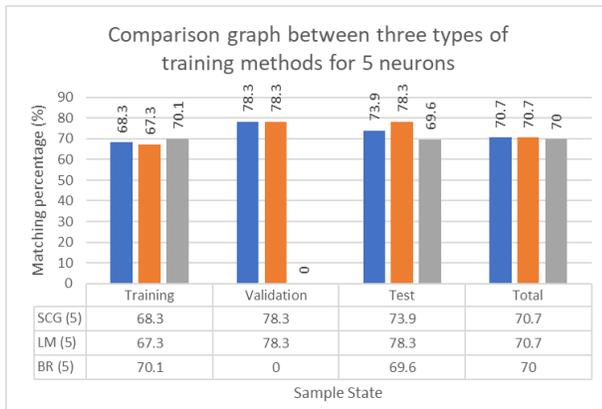
Amount of data	Number of neurons	Sample State Matching percentage (%)			
		Training	Validation	Test	Total
109	5	62.3	43.8	75.0	61.5
	10	67.5	68.8	62.5	67.0
	15	62.3	56.3	56.3	60.6
	20	48.1	50.0	25.0	45.0
150	5	66.3	73.9	73.9	68.7
	10	74.0	52.2	73.9	70.7
	15	61.5	65.2	78.3	64.7
	20	64.4	73.9	56.6	64.7

Consequently, the training using the three methods was executed on the 150 data sets. It can be observed from Table 5 that for 10 neurons, SCG was able to achieve 72.7% matching percentage compared to other training methods where LM and BR have performed with 71.3% and 68.7% accuracy, respectively.

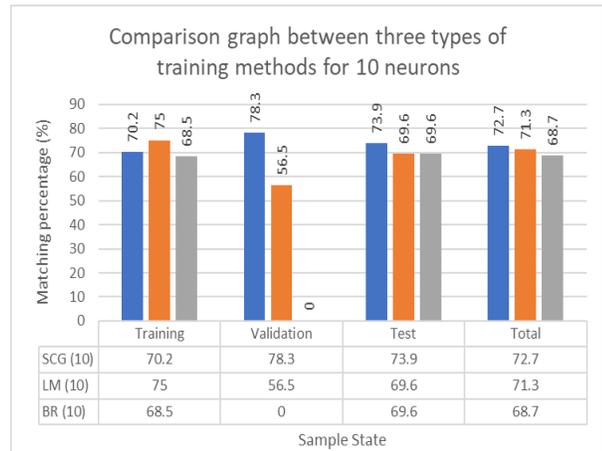
TABLE V
COMPARISON OF DIFFERENT TYPES OF TRAINING METHODS

Training method	Number of neurons	Sample State Matching percentage (%)			
		Training	Validation	Test	Total
SCG	5	68.3	78.3	73.9	70.7
	10	70.2	78.3	73.9	72.7
LM	5	67.3	78.3	78.3	70.7
	10	75.0	56.5	69.6	71.3
BR	5	70.1	N/A	69.6	70.0
	10	68.5	N/A	69.6	68.7

Fig. 13 shows the graph for Table 5 that is plotted for better visualization where Fig. 13(a) is plotted for 5 neurons and Fig. 13(b) for 10 neurons. It can be observed that the differences for each method for All state are between ± 0.7 for 5 neurons and ± 4 for 10 neurons. Thus, it can be concluded that with the highest accuracy of matching percentage, SCG is the most suitable training method.



(a): Comparison graph between SCG, LM and BR for 5 neurons



(b): Comparison graph between SCG, LM, and BR for 10 neurons

Fig. 13: Comparison graph for three training methods

V. CONCLUSION

In conclusion, the texture recognition by using the three-fingered robot hand 6-axis force/torque sensor with position-based impedance control was done where it was able to differentiate between two types of object textures. This was done by analyzing the F_{rate} value using statistical methods which are minimum, maximum, and mean of the data. Besides, sufficient amount of data is needed for ANN training for better accuracy performance where it has been proven in simulation 1 that 150 data gave 70.7% accuracy compared to 67% by the 109 data. From simulation 2, it can be concluded that the most suitable training method for the ANN is SCG which has produced 72.7% accuracy performance compared to LM with 71.3% and BR 68.7%. The selection of K_d value can allow the robot hand to have accurate measurement when differentiating hard and soft object textures. The findings from this work can be used for real-time execution of the control in the next study.

ACKNOWLEDGEMENT

This research is fully supported by FRGS (600-IRMI/FRGS 5/3 (326/2019) grant. The authors acknowledged Ministry of Higher Education (MOHE) for the approved fund and to Universiti Teknologi MARA for providing the laboratory space and equipment.

REFERENCES

- [1] Z. Zhao, W. Shang, H. He, and Z. Li, "Grasp prediction and evaluation of multi-fingered dexterous hands using deep learning," *Rob. Auton. Syst.*, vol. 129, p. 103550, 2020.
- [2] K. Rani and N. Kumar, "Design of Intelligent Hybrid Force and Position Control of Robot Manipulator," *Procedia Comput. Sci.*, vol. 125, pp. 42–49, 2018.
- [3] Z. Yang, J. Peng, and Y. Liu, "Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer," *Neurocomputing*, vol. 331, pp. 263–280, 2019.
- [4] W. He and Y. Dong, "Adaptive Fuzzy Neural Network Control for a Constrained Robot Using Impedance Learning," vol. 29, no. 4, pp. 1174–1186, 2018.
- [5] S. Rispaal, A. K. Rana, and V. Duchaine, "Textures recognition through tactile exploration for robotic applications," 2017 3rd Int. Conf. Control. Autom. Robot. ICCAR 2017, pp. 121–128, 2017.
- [6] C. C. Beltran-Hernandez *et al.*, "Learning Force Control for Contact-Rich Manipulation Tasks With Rigid Position-Controlled Robots," in *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5709–5716, Oct. 2020.
- [7] T. Tsuji, K. Sato, and S. Sakaino, "Contact Feature Recognition Based on MFCC of Force Signals," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5153–5158, 2021.
- [8] T. Sun, J. Back, and H. Liu, "Combining Contact Forces and Geometry to Recognize Objects during Surface Haptic Exploration," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2509–2514, 2018.
- [9] M. Kaboli and G. Cheng, "Robust Tactile Descriptors for Discriminating Objects from Textural Properties via Artificial Robotic Skin," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 985–1003, 2018.
- [10] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*. PWS Publishing Co., 1997.
- [11] D. J. C. MacKay, "Bayesian Interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, May 1992.
- [12] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, Jan. 1993.
- [13] K. Nasir, R.L.A. Shauri, N.M. Salleh, N.H. Remeli, "Kinematic modeling for PID position control of a new three-fingered robotic hand," *ARPJ. J. of Engineering and Applied Sciences*, vol. 14, No.2, pp. 335–343, 2019.
- [14] R. L. A. Shauri, A. B. Roslan, and Z. S. M. Zarib, "Application of 6-axis force torque (F/T) sensor for a three fingered robot hand impedance control," in *2020 IEEE 10th International Conference on System Engineering and Technology, ICSET 2020 - Proceedings, 2020*, pp. 262–266.
- [15] K. Nasir, R. L. A. Shauri, N. M. Salleh, and N. H. Remeli, "Effects of varying impedance parameters for position-based impedance control of a three-fingered robot," *Pertanika J. Sci. Technol.*, vol. 25, no. S7, pp. 111–122, 2017.