Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Online Terrain Classification Using Neural Network for Disaster Robot Application

Muhammad Anwar Sanusi[1], Bima Sena Bayu Dewantara[2], Setiawardhana[3], Riyanto Sigit[4]
[1]anwarsanusi0@gmail.com, [2]bima@pens.ac.id, [3]setia@pens.ac.id, [4]riyanto@pens.ac.id
Politeknik Elektronika Negeri Surabaya

Article Information

Submitted : 3 Jan 2023
Reviewed: 11 Jan 2023
Accepted : 26 Feb 2023

Keywords

Terrain Classification, Disaster Robot, Neural Network, and Inertial Measurement Unit

Abstract

A disaster robot is used for crucial rescue, observation, and exploration missions. In the case of implementing disaster robots in bad environmental situations, the robot must be equipped with appropriate sensors and good algorithms to carry out the expected movements. In this study, a neural network-based terrain classification that is applied to Raspberry using the IMU sensor as input is developed. Relatively low computational requirements can reduce the power needed to run terrain classification. By comparing data from the Accelerometer, Gyroscope, and combined Accelero-Gyro using the same neural network architecture, the tests were carried out in a not moving position, indoors, on asphalt, loose gravel, grass, and hard ground. In its implementation, the mobile robot runs over the field at a speed of about 0,5 m/s and produces predictive data every 1,12s. The prediction results for online terrain classification are above 93% for each input tested.

A. Introduction

The Indonesian archipelago is located on the "Ring of Fire," the confluence of three tectonic plates in the world, namely the Australian, Pacific, and Eurasian [1]. This area has a high potential for earthquakes, tsunamis, volcanic eruptions, and landslides [2]. The Meteorology, Climatology, and Geophysics Agency (BMKG) confirmed that Kalimantan Island is the only island in Indonesia with the lowest level of seismic activity [3].

From active tectonic and volcanic activity data, it often occurs in several regions in Indonesia. This is evidenced by the fact that Indonesia's regions often have earthquakes and volcanic eruptions annually [3]. In this case, the impact caused by the earthquake was quite extensive and massive, such as the destruction of facilities and property and the loss of people's lives [2][4]. Active volcanoes also risk emitting toxic gases, hot clouds, cold lava, and lava [5]. The impact caused by lava is that it destroys everything in its path, and also, no living things are safe when a disaster occurs [4]. Efforts are needed so that areas hit by disasters can be evacuated immediately to ensure the safety of the affected population.

Searching for victims, assessing the damaged situation, and planning rescue and recovery are important issues [1][6]. Uncertainty, danger, and limited availability of rescuers all affect the ability to save lives. Required robots or unmanned systems can be operated in disaster areas to help search for victims. Robot-type UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) are very often used for search and rescue missions [7]. UAVs are used for area mapping missions because they can roam through the air. The UGV has a good ground cruising capability and is suitable for exploration and observation missions.

Disaster robots usually still use manual control via teleoperated mode [7]. The user controls the robot remotely by using the visualization from the FPV camera to navigate the robot. This camera will not function optimally when faced with foggy, smoky conditions, heavy rain, dust, excess, or lack of light intensity. When the robot encounters a certain field, when it cannot be seen visually by the user, it will cause new problems. In the implementation, disaster robots need to be equipped with several supporting sensors such as GPS, Lidar, and IMU to carry out autonomous missions [8][9].

In this case, researchers only focus on applying terrain classification algorithms to help disaster robot navigation. In several studies related to other classification terrains, many of them are implemented using several sound inputs [10], visual inputs [11], and also vibrations [12][13][14]. Of course, with different methods and different testing schemes. In implementing terrain classification cases, they use an offline classification or online classification approach [15][16][17].

In the application of the robotic field, most of them use non-holonomic based. This control scheme is relatively easy and reliable when applied to fields that cannot be traversed by holonomic based. The most accessible and reliable application is the DDMR system [8][18][19]. In this study, researchers tested using inexpensive devices. The prototype disaster robot is based on a 4-wheels differential-driven mobile robot (DDMR) equipped with an Inertial Measurement Unit (IMU) sensor.

The IMU used is BNO055 which can output accelerometer, gyroscope, and magnetometer data. In this case, the researcher only uses data from the accelerometer and gyroscope sensors to record vibrations on the robot when it crosses the terrain. Researchers use a neural network implemented on the Raspberry Pi 4B to process vibration data in real-time. Using the same neural network architecture, researchers will compare the predictions of each input to be tested. The data used as input are the gyroscope, the accelerometer, and the gyro-accelerometer combination. Terrain classification testing will be carried out in real-time, which is called online terrain classification. The purpose of online terrain classification is to input from the actual robot disaster navigation system. The prediction results are expected to help the robot to determine the path that the robot can still pass. The robot will also be able to adjust the speed based on the type of terrain it travels to be more efficient, effective, and safe when carrying out observation and exploration missions.

B. Research Method

The research implementation can be described in the system design. The system design describes the entire process contained in the system, which can be shown in the following Figure 1.

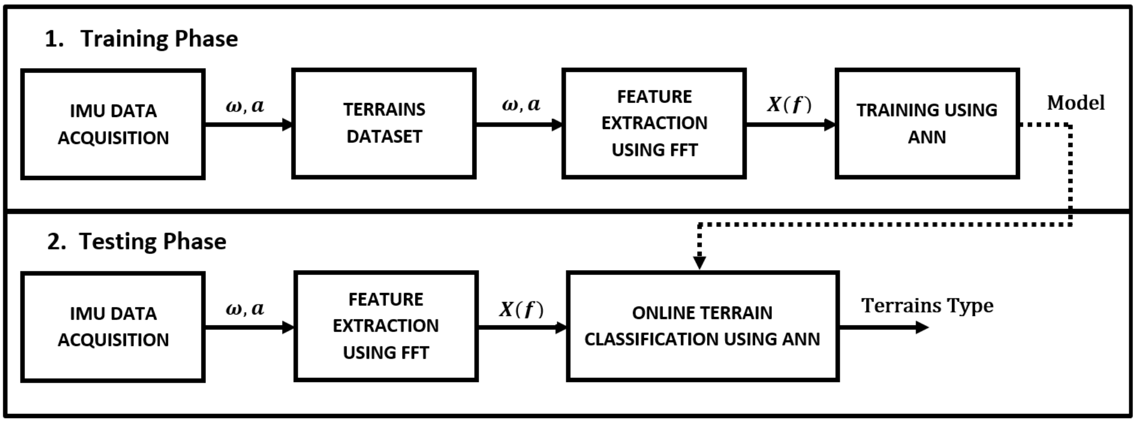

Figure 1. System Design

Based on the design system in Figure 1, the explanation for each step is as follows:

1. Training Phase

1.1 IMU Data Acquisition (Training)

A disaster robot prototype based on a 4-wheeled mobile robot, seen in Figure 2, was created to collect IMU data. This mobile robot has four wheels and four driving motors (PG36 DC Motor). Adapting from a non-holonomic based robot system with a DDMR (Differential Driven Mobile Robot) drive system. The size of the mobile robot is 374,5 x 608 x 277,7 mm. As for the serrated wheel, it has a diameter of 158 mm and a thickness of 115 mm. Meanwhile, the weight of each wheel is 525 grams, and the total weight is 4.75 kg.

This robot uses the Arduino Mega Board to accommodate the entire system. A BTN7960 motor driver drives each PG36 24V DC Motor. To drive this robot, connect the Bluetooth HC-05 to the Bluetooth RC Controller application on the Play Store. The robot is powered by 4 Lion 3.7V 3000mAH batteries installed in series. Mobile robot is equipped with a 9-DOF Inertial Measurement Unit (IMU), which is the IMU BNO055 turning the sensor data from an accelerometer, gyroscope, and magnetometer into actual "3D space orientation". As for data from IMU BNO055, it is read by Arduino Uno, which will later send the data serially to the Raspberry Pi 4B. A 5V 5A DC-DC Buck Converter powers this Raspberry. The power is obtained from 4 lion 3.7v 3000mAh batteries installed in series.

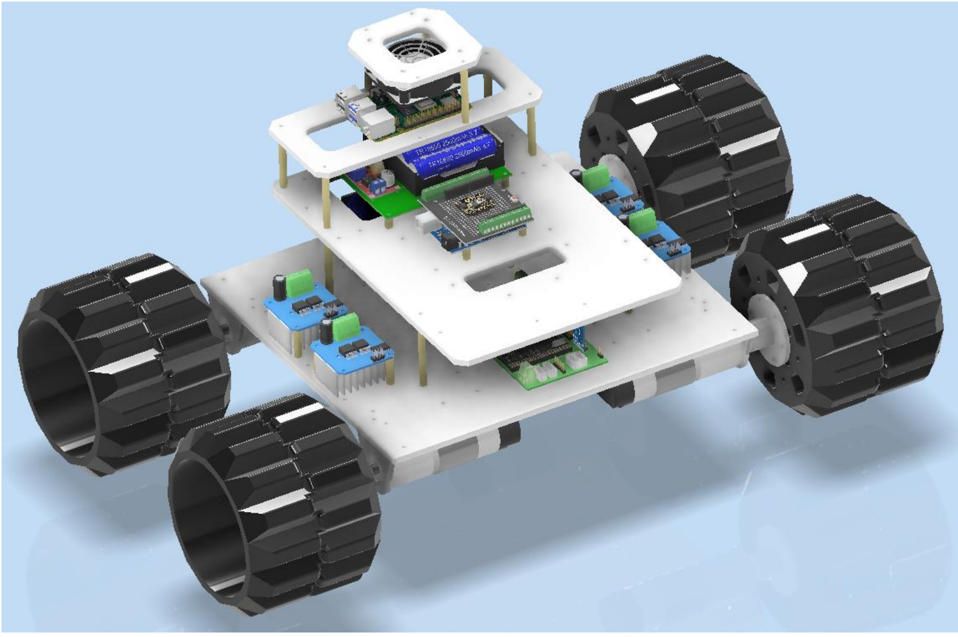

Figure 2. Prototype disaster robot design

As for the IMU position, it is on the top layer and right in the middle of the robot body. The IMU BNO055 has the following outputs: Absolute Orientation (Euler Vector, 100Hz), Absolute Orientation (Quaternion, 100Hz), Angular Velocity Vector (rad/s, 100Hz), Acceleration Vector (m/s^2, 100Hz), Magnetic Field Strength Vector (uT, 20Hz), Linear Acceleration Vector (m/s^2, 100Hz), Gravity Vector (m/s^2, 100Hz) and Temperature (Celsius, 1Hz). However, in this study, we only used accelerometer and gyroscope data.

Data acquisition was done on five different terrains and stop-moving condition, which can be seen in Figure 3, and are the following:

Figure 3. Multiple terrains for IMU data acquisition

As for raw data samples, each is visualized in the graph below:

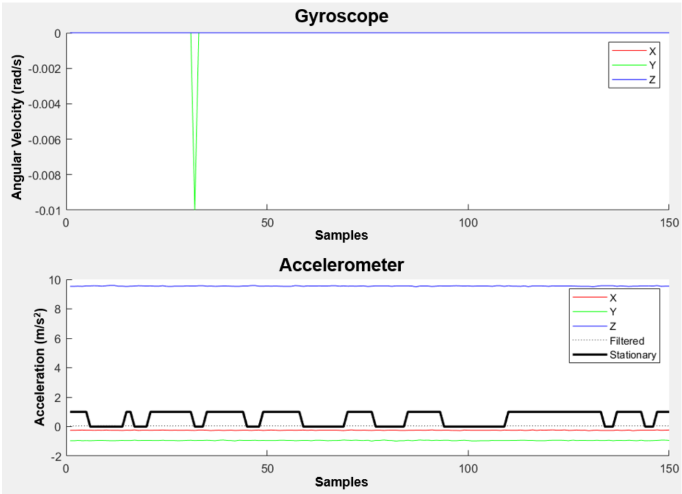

Figure 4. Gyroscope and accelerometer data visualization for stop-moving conditions

The graph in Figure 4 is a sample of raw data for stop-moving conditions. Ideally, for gyroscope data, the x-axis, y-axis, and z-axis are 0 rad/s, but in reality, some anomalies are read. In accelerometer data, the x-axis and y-axis are near 0. The z-axis is affected by gravity, so it is measured at 9,8 m/s2.

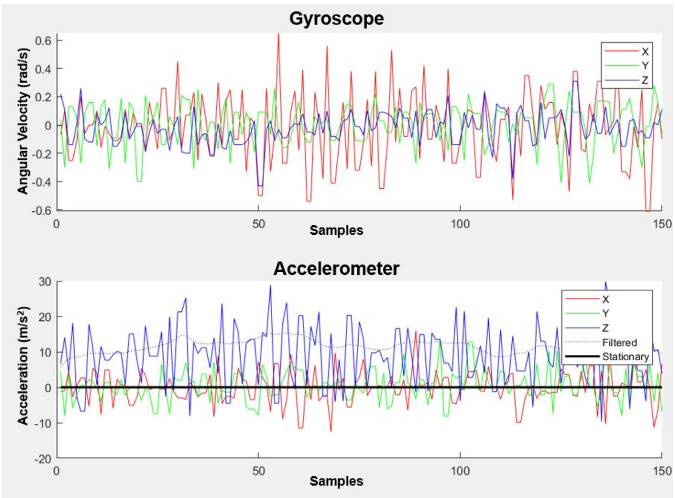

Figure 5. Gyroscope and accelerometer data visualization for indoor floor terrain

The graph in Figure 5 is a sample of raw data for indoor floor terrain. In this terrain test, the vibration is not too large. The characteristic of this terrain is that the surface is flat and quite slippery. So for gyroscope data, the y-axis and z-axis are in the amplitude range of -0.2 to 0.2. Meanwhile, the x-axis range is in the range above it. In the accelerometer data, the z-axis has the largest amplitude range from 0 to 20. The average amplitude range for the x and z axes is -8 to 8.

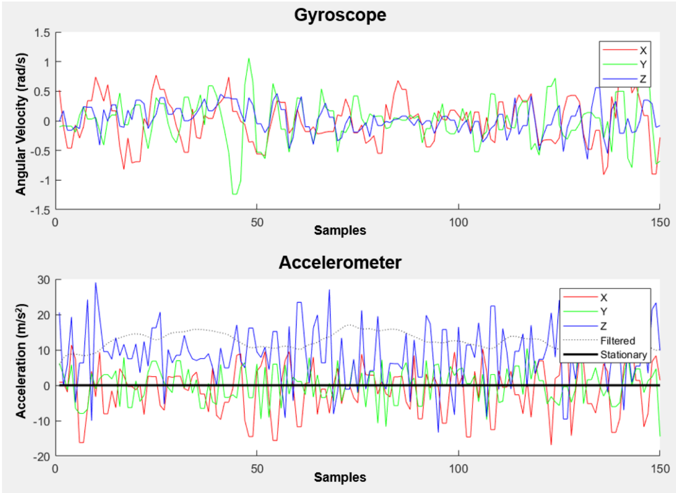

Figure 6. Gyroscope and accelerometer data visualization for asphalt terrain

The graph in Figure 6 is a sample of raw data for asphalt terrain. A distinctive feature of this terrain is that the surface is uneven and quite rough. So for gyroscope data, the x, y, and z axes are, on average, in the amplitude range of -0.5 to 0.5. In accelerometer data, the average z-axis has an amplitude range from -10 to 25. The average amplitude range for the x and z axes is -10 to 10.

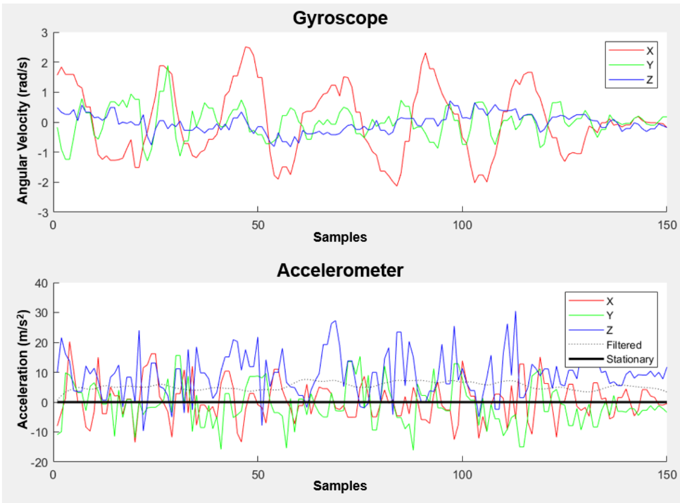

Figure 7. Gyroscope and accelerometer data visualization for loose gravel terrain

The graph in Figure 7 is a sample of raw data for loose gravel terrain. The characteristics of this terrain are that the surface is uneven and slippery, and the rocks wobble quickly when the robot passes. So for gyroscope data, the z-axis is in the amplitude range of -0.5 to 0.5. while the y-axis is in the range of -1 to 1, and the x-axis forms a large wave pattern. The amplitude range is in the range of -2 to 2. In accelerometer data, the average z-axis has an amplitude range from -5 to 20. The average amplitude range for the x and z axes is -10 to 10.

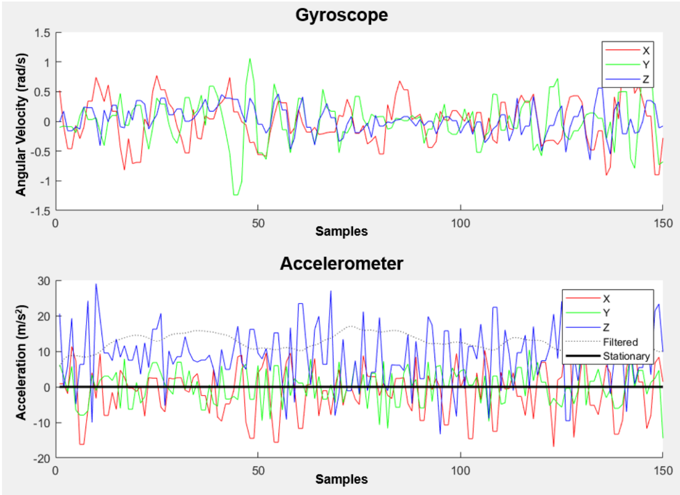

Figure 8. Gyroscope and accelerometer data visualization for grass terrain

The graph in Figure 8 is a sample of raw data for grass terrain. The characteristic feature of this field is that the surface is relatively flat and the surface is soft enough to dampen vibrations. However, in some areas, there are rocks and bare grass. So for gyroscope data, the average z-axis is in the amplitude range of -0.2 to 0.2. Meanwhile, the x and y axes are in the amplitude range of -0.5 to 0.5. In accelerometer data, the average z-axis has an amplitude range from 5 to 15. The average amplitude range for the x and z axes is -5 to 5.

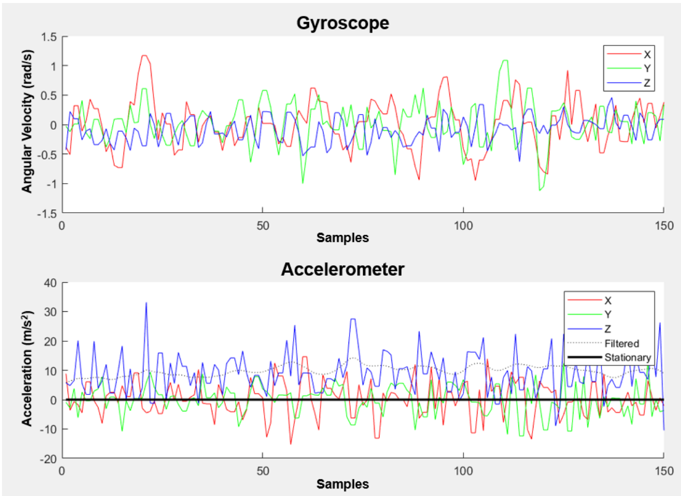

Figure 9. Gyroscope and accelerometer data visualization for soil terrain

The graph in Figure 9 is a sample of raw data for soil terrain. A distinctive feature of this terrain is that its surface is relatively flat and the surface is quite flat. However, in some areas, there are rocks. So for gyroscope data, the average z-axis is in the amplitude range of -0.4 to 0.4. In contrast, the x and y axes are in the amplitude range of -1 to 1. In accelerometer data, the z-axis has an average amplitude range from 0 to 25. The average amplitude range for the x and z axes is -10 to 10.

1.2 Terrains Dataset

For saving terrains dataset, we use python and pyserial libraries. Which we then save in a file with the extension (.csv). Retrieval of datasets is carried out on each terrain as much as 60x. For every 1 dataset, 150 accelerometer and gyroscope data samples are obtained. The specifications of the IMU are 100Hz for the accelerometer and gyroscope, so in 1 dataset, it takes 1,5 seconds. The average speed of the robot when passing through the terrain is 0,5 m/s.

```
Sample,Gyro_x,Gyro_y,Gyro_z,Accelero_x,Accelero_y,Accelero_z
0,0.88,0.31,0.56,-7.97,-10.85,10.01
1,1.57,-0.17,0.48,-7.97,-10.85,10.01
2,1.84,-0.90,0.33,-3.42,-9.93,21.54
3,1.59,-1.24,0.26,1.35,9.80,16.04
4,1.59,-1.24,0.26,20.21,9.09,13.70
5,1.59,-0.51,0.42,10.68,6.50,3.83
6,1.23,0.27,0.05,0.21,3.17,3.53
7,1.14,0.78,0.57,0.21,3.17,3.53
8,0.51,0.31,0.33,-8.35,-2.85,12.08
9,0.51,0.31,0.33,-9.73,5.25,9.40
10,-0.35,-0.44,0.15,-3.22,6.50,11.34
11,-1.08,-0.57,0.17,15.00,3.63,7.34
12,-1.24,-0.22,0.06,-3.85,-0.66,0.67
13,-1.12,0.36,0.16,-3.85,-0.66,0.67
14,-1.27,0.68,0.49,5.08,2.97,8.33
15,-1.27,0.68,0.49,2.09,-11.48,9.66
16,-1.25,0.54,-0.07,2.22,4.02,12.41
17,-1.20,0.74,0.00,-4.08,-5.14,-1.69
18,-0.88,0.49,-0.04,1.05,0.90,8.36
```

Figure 10. Screenshot of loose gravel terrain raw data

Figure 10 shows a snapshot of the raw loose gravel terrain data. The vibrations when the robot crosses the field are stored as a dataset. The data stored is in the form of sample sequences, gyroscope data (x-axis, y-axis, z-axis), and accelerometer data (x-axis, y-axis, z-axis).

1.3 Feature Extraction Using FFT

Feature extraction is helpful in identifying the most discriminatory characteristics in a signal [20]. It is recommended for feature extraction so that it is easier to use by machine learning or deep learning algorithms. The use of raw data often gives poor results due to high data rates and information redundancy when used for machine learning. The Fast Fourier Transform (FFT) algorithm transforms raw data (accelerometer and gyroscope) in the time domain into the frequency domain. The FFT formula that we use is as follows:

$$\mathrm{X(f)} = \sum x_k . e^{-il\Delta\omega k} \tag{1}$$

$$= \sum x_k . e^{-il2\pi f k} \tag{2}$$

$$= \sum x_k . e^{-ilk2\pi/_N} \tag{3}$$

Where,
X is the value of the transformation results in the frequency domain.
f is frequency.
x_k is sampling values to (-k) of the time domain.
l is discrete frequency index.
k is the time index.
k is the time index.

1.4 Training Using ANN

Multi-Layer Perceptron classifiers, which feedforwards Artificial Neural Networks (ANNs), were applied in this research. These classifiers proved to apply to online classification tasks on similar projects. As for the dataset that we use for training the neural network model, it is shown in Table 1:

Table 1. Terrain dataset

	Types of Terrains						Total
	Not Moving	**Indoor Floor**	**Asphalt**	**Loose Gravel**	**Grass**	**Soil**	
Dataset	7500	9000	9000	9000	9000	9000	52500

The dataset obtained is 52500 for the total of the entire test. 80% of the dataset is used as training data, and the remaining 20% is used as testing data. As for training the NN model using Visual Studio Code software, we use several extensions and libraries such as Jupyter, Python 3.7.6, Tensorflow 2.1.0, NumPy, pandas, and matplotlib. The neural network architecture that we use is as follows:

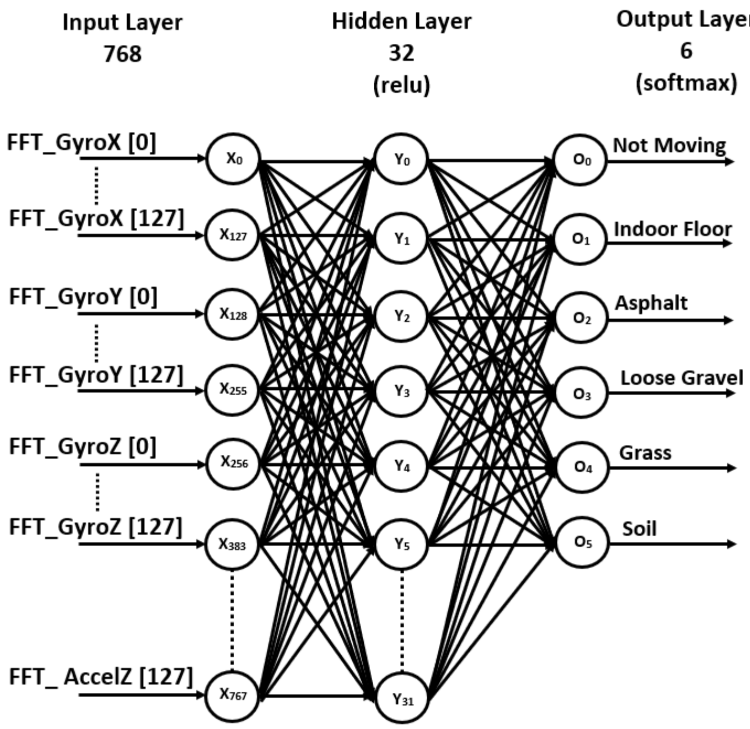

Figure 11. Neural network architecture for accelerometer-gyroscope input

The terrain classification architecture using accelero-gyro input is shown in Figure 11. The data that enters this ANN is fourier data that has been transformed with FFT. The total input is 768 (384 from gyroscope FFT data and 384 from accelerometer FFT data). So each axis gives each 128 FFT data. There are 32 hidden layers, and there are six classification outputs. The architecture above combines two inputs, so for training models that use only a gyroscope or accelerometer, only 384 data is needed from one of the sensor inputs. So windowing for gyroscope or accelerometer input uses 384 FFT data. Meanwhile, the combine (gyro-accelero) uses 786 FFT data.

2. Testing Phase

The testing phase contains several similar steps, such as IMU data acquisition and feature extraction using FFT. The IMU data acquisition is slightly different from the training phase. In this phase, raw data from IMU will be sent directly to Raspberry and transformed using FFT. So that without the process of storing data earlier, it will be the input of the ANN. The ANN model is obtained from the training results for experimental testing. The online terrain classification process is executed in real-time from IMU sensor data.

The robot will be tested on indoor floor terrain, asphalt, loose gravel, grass, and soil when the robot stops moving. The online terrain classification program is executed, and the robot will be remotely forwarded straight through the testing terrain. The robot will be tested when the situation stops in each terrain in the stop-moving test. As for after testing, an evaluation will be carried out regarding the predicted results obtained.

C. Result and Discussion

As for this research, online terrain classification and experimental tests were carried out on a mobile robot. The hardware specifications used in this experimental test can be seen in Table 2.

Table 2. Hardware specifications used for experimental testing

Hardware	Description
Laptop	Lenovo Legion 5 15ITH6H
Processor	Intel(R) Core(TM) i7-11800H CPU @ 4.60GHz
RAM	2x 8GB SO-DIMM DDR4-3200
HDD	512GB SSD M.2 2280 PCIe 4.0x4 NVMe
Xiaomi Pocophone F1	Wifi Hotspot
Raspberry Pi 4B	SBC/Mini PC for online terrain classification
Arduino Uno	Microcontroller for reading IMU data
Adafruit BNO055	IMU sensor
OS and Software used in experimental testing	Windows 11 Home VNC Viewer Raspberry Pi OS (Legacy) Python 3.7.6 Tensorflow 2.1.0

In the online terrain classification experimental test, Raspberry is first turned on. In this case, the position of Raspberry is mounted on the robot. The Raspberry runs headless with the Raspberry interface displayed on the laptop. This process requires the same network and hotspot connection connected to the Raspberry and the laptop. Then the online terrain classification program is run on the Raspberry, and the robot will run forward on each terrain. This test compares predictions from online terrain classification using different inputs on the same neural network architecture. The inputs being compared are the accelerometer, the gyroscope, and the accelero-gyro combination.

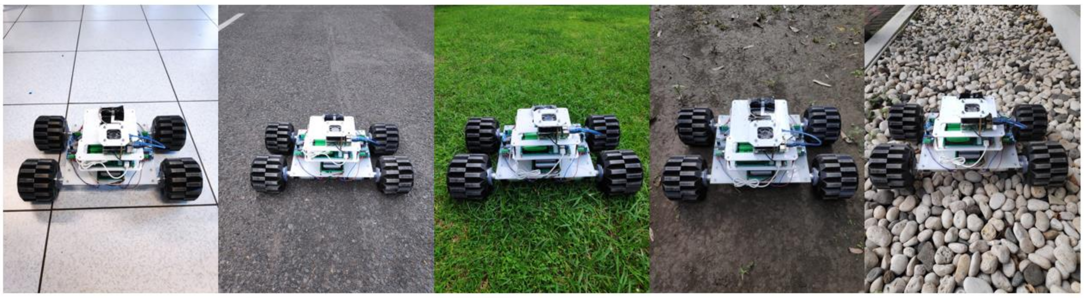

Figure 12. Online terrain classification testing on several terrains

Figure 12 shows the online terrain classification test on several terrains. The tests were carried out on the surface of the indoor floor, asphalt, grass, soil/ground, loose gravel, and stopped/not moving. Each terrain will be tested with accelerometer input, the gyroscope, and the accelero-gyro combination. Each online terrain classification test on each terrain will take 30 prediction data. Six prediction results are taken for each terrain for testing, not moving conditions.

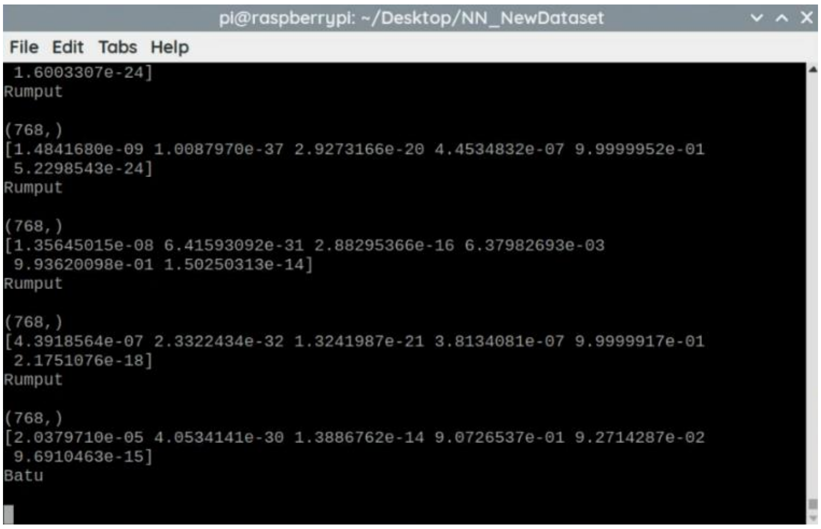

Figure 13. Online terrain classification prediction on grass terrain

Figure 13 shows the online terrain classification prediction on grass terrain. The terrain classification program is stored on raspberry. Then run it on the raspberry os terminal. This program runs using Python 3.7.6, Tensorflow 2.1.0, and several required libraries. The photo above is just a snapshot of the entire test. Regarding the overall test results are shown in the tables below.

Table 3. Online terrain classification testing using accelerometer as input

		Terrain Input (Accelerometer)					
		Not Moving	Indoor Floor	Asphalt	Loose Gravel	Grass	Soil
Predicted Terrains	Not Moving	**30 (100%)**					
	Indoor Floor		**27 (90%)**				2 (6,66%)
	Asphalt		3 (9,99%)	**28 (93,3%)**	1 (3,33%)		
	Loose Gravel				**29 (96,6%)**	1 (3,33%)	1 (3,33%)
	Grass					**28 (93,3%)**	
	Soil			2 (6,66%)		1 (3,33%)	**27 (90%)**

Table 3 shows the results of online terrain classification testing using the accelerometer as input. A pretty good prediction is obtained by using 384 accelerometer sample data (128 data for each axis) for every 1x prediction. It takes an interval of 1,12s for each prediction generated in this test. The best prediction is when the robot is not moving (100% accuracy) and loose gravel terrain (96,6% accuracy). In contrast, predictions for indoor floor terrain and soil/ground terrain have the worst accuracy, namely 90%.

Table 4. Online terrain classification testing using gyroscope as input

		Terrain Input (Gyroscope)					
		Not Moving	Indoor Floor	Asphalt	Loose Gravel	Grass	Soil
Predicted Terrains	Not Moving	**30 (100%)**					
	Indoor Floor		**28 (93,3%)**				
	Asphalt		2 (6,66%)	**29 (96,6%)**	1 (3,33%)		2 (6,66%)
	Loose Gravel				**29 (96,6%)**		
	Grass					**30 (100%)**	
	Soil			1 (3,33%)			**28 (93,3%)**

Table 4 shows the results of online terrain classification testing using the gyroscope as input. A pretty good prediction is obtained by using 384 gyro sample data (128 data for each axis) for every 1x prediction. It takes an interval of 1,12s for each prediction generated in this test. The best prediction is when the robot is not moving (100% accuracy) and on grass terrain (100% accuracy). In contrast, predictions for indoor floor terrain and ground terrain have the worst accuracy, namely 93,3%.

Table 5. Online terrain classification testing using accelero-gyro as input

Predicted Terrains		Terrain Input (Accelero-Gyro)					
		Not Moving	Indoor Floor	Asphalt	Loose Gravel	Grass	Soil
	Not Moving	**30 (100%)**					
	Indoor Floor		**29 (96,6%)**				
	Asphalt		1 (3,33%)	**27 (90%)**			1 (3,33%)
	Loose Gravel			1 (3,33%)	**30 (100%)**	3 (9,99%)	1 (3,33%)
	Grass					**27 (90%)**	
	Soil			2 (6,66%)			**28 (93,3%)**

Table 5 shows the results of online terrain classification testing using the accelero-gyro as input. A pretty good prediction is obtained using 384 accelerometer sample data (128 data for each axis) and 384 gyroscope sample data (128 data for each axis) for every 1x prediction. It takes an interval of 1,12s for each prediction generated in this test. The best prediction is when the robot is not moving (100% accuracy) and loose gravel terrain (100% accuracy). In contrast, predictions for asphalt terrain and grass terrain have the worst accuracy, namely 90%.

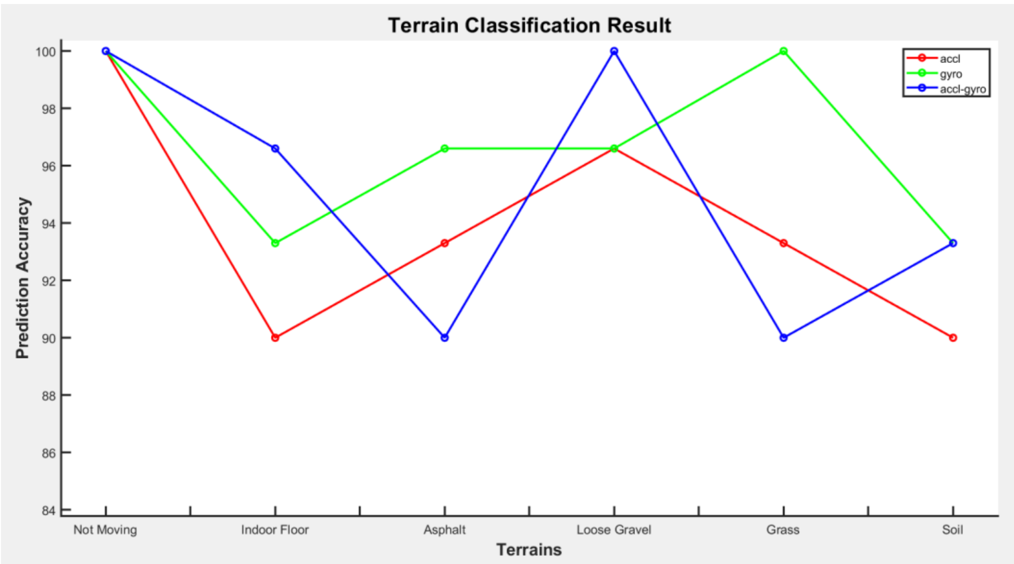

Figure 14. Comparison chart of online terrain classification using neural network under different inputs.

Figure 14 shows the comparison chart of online terrain classification under different inputs. Of all the input data tested, the time interval needed to generate predictions is the same, namely 1,12s for each prediction. The average accuracy for each input is the accelerometer with an accuracy of 93.86%, gyroscope with an accuracy of 96.63%, and accelero-gyro with an accuracy of 94.98%. The gyroscope obtains the best results as input.

D. Conclusion

This study proposes an online terrain classification using a neural network vibration-based method for disaster robots. Compared to existing terrain classification studies, using an IMU sensor does not require a high-spec PC and high power. It is more resistant to disturbances in the outdoor environment (e.g., dust, light intensity, and fog). The average time for the system to run terrain classification online using a neural network is estimated to be 1,12s. From the input data that has been tested (Accelerometer, Gyroscope, and Accelerometer-Gyroscope), the best accuracy is obtained by the Gyro input (accuracy of 96,63%). These results are considered sufficient to be applied to the disaster robot for the actual mission. In future research, we will apply terrain classification to better disaster robots. The current robot is not robust enough when applied to a real mission. The researcher will also compare terrain classification using other time series forecasting algorithms such as Autoregressive Integrated Moving Average (ARIMA) and Long Short Term Memory (LSTM).

E. Acknowledgment

Acknowledgments to the Department of Research and Community Service (DRPM DIKTI) of Indonesia under grant number: 016/PL14/PG.1/SP2P/2021 for funding this research. We would also like to show our gratitude to all researchers and friends involved in this research and to facilities support from Politeknik Elektronika Negeri Surabaya (PENS), so we can complete our research.

F. References

[1] S. Kuswadi and R. Sigit, “Robot Kebencanaan–Latar Belakang, Perancangan dan Realisasinya,” in *Seminar Nasional Humaniora dan Aplikasi Teknologi Informasi (SEHATI)*, 2020, vol. 6, no. 1, pp. 8–16.

[2] M. Cassidy and L. Mani, “Huge volcanic eruptions: Time to prepare.” Nature Publishing Group, 2022.

[3] Irsyam et al, *Gempa bumi di Indonesia*. Katalog Pusat Studi Gempa Bumi Nasional (PuSGeN), 2016.

[4] R. Rahayu, D. P. Ariyanto, K. Komariah, S. Hartati, J. Syamsiyah, and W. S. Dewi, “Dampak erupsi Gunung Merapi terhadap lahan dan upaya-upaya pemulihannya,” *Caraka Tani: Journal of Sustainable Agriculture*, vol. 29, no. 1, pp. 61–72, 2014.

[5] Endah, “Semburan Gas Beracun Kawah Ijen,” *https://geologi.co.id/2018/03/25/semburan-gas-beracun-kawah-ijen*, Mar. 25, 2018.

[6] A. N. G. O. Son Kuswadi, M. N. Tamara, and S. Indra Adji, “Optimasi Sistem Navigasi Robot Bencana dengan Algoritma Bug dan Jaringan Syaraf Tiruan,” *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, vol. 5, no. 5, 2018.

[7] K. Nagatani *et al.*, “Volcanic ash observation in active volcano areas using teleoperated mobile robots-Introduction to our robotic-volcano-observation

project and field experiments," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013, pp. 1–6.

[8] J. Iqbal, R. Xu, S. Sun, and C. Li, "Simulation of an autonomous mobile robot for LiDAR-based in-field phenotyping and navigation," *Robotics*, vol. 9, no. 2, p. 46, 2020.

[9] E. Cetin, C. Barrado, G. Munoz, M. Macias, and E. Pastor, "Drone navigation and avoidance of obstacles through deep reinforcement learning," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–7.

[10] N. Riopelle, P. Caspers, and D. Sofge, "Terrain classification for autonomous vehicles using bat-inspired echolocation," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–6.

[11] W. Wang *et al.*, "A visual terrain classification method for mobile robots' navigation based on convolutional neural network and support vector machine," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 4, pp. 744–753, 2022.

[12] F. Lomio, E. Skenderi, D. Mohamadi, J. Collin, R. Ghabcheloo, and H. Huttunen, "Surface type classification for autonomous robot indoor navigation," *arXiv preprint arXiv:1905.00252*, 2019.

[13] D. Csík, Á. Odry, J. Sárosi, and P. Sarcevic, "Inertial sensor-based outdoor terrain classification for wheeled mobile robots," in *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)*, 2021, pp. 159–164.

[14] M. Wang, L. Ye, and X. Sun, "Adaptive online terrain classification method for mobile robot based on vibration signals," *Int J Adv Robot Syst*, vol. 18, no. 6, p. 17298814211062036, 2021.

[15] W. Shi, Z. Li, W. Lv, Y. Wu, J. Chang, and X. Li, "Laplacian support vector machine for vibration-based robotic terrain classification," *Electronics (Basel)*, vol. 9, no. 3, p. 513, 2020.

[16] M. Mei, J. Chang, Y. Li, Z. Li, X. Li, and W. Lv, "Comparative study of different methods in vibration-based terrain classification for wheeled robots with shock absorbers," *Sensors*, vol. 19, no. 5, p. 1137, 2019.

[17] S. Otte, C. Weiss, T. Scherer, and A. Zell, "Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5603–5608.

[18] S. Khatoon, M. Istiyaque, S. A. Wani, and M. Shahid, "Design Kinematics and Control for a Differential Drive Mobile Robot," in *Renewable Power for Sustainable Growth*, Springer, 2021, pp. 189–196.

[19] A. Ismaiel and M. Y. Filimonov, "A simulation-based study to calculate all the possible trajectories of differential drive mobile robot," in *AIP Conference Proceedings*, 2021, vol. 2333, no. 1, p. 070008.

[20] P. Sarcevic, S. Pletl, and Z. Kincses, "Comparison of time-and frequency-domain features for movement classification using data from wrist-worn sensors," in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 2017, pp. 261–266.