Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi
Vol.8 No.2 July 2023, P-ISSN : 2502-3470, E-ISSN : 2581-0367

Design and Implementation of Artificial Intelligence-based Real-time Pothole Detection for IoT Smart Infrastructure

Rahardhita Widyatra Sudibyo[1], Akhmad Alimudin[2]

[1]*Department of Electronic Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia*

[2]*Department of Multimedia Creative Technology, Politeknik Elektronika Negeri Surabaya, Indonesia*

[1]widi@pens.ac.id (*)

[2]alioke@pens.ac.id

Received: 2023-04-27; Accepted: 2023-05-30; Published: 2023-06-05

Abstract— The Internet of Things (IoT) has been extensively deployed for Smart Cities due to its ability to process many different and heterogeneous end systems. The IoT innovation encourages artificial intelligence applications to process data. In Smart City infrastructure, the road is a critical component of transportation infrastructure that supports the economic, social, and cultural things of community life and various aspects of community life. Road conditions affect a variety of community activities. Good roads enhance comfort and support local businesses. However, many roads remain in bad condition, such as potholes. Various methods have been attempted to identify potholes, especially the two-dimensional imaging method. This paper proposes the real-time Artificial Intelligence detection of potholes using the Convolutional Neural Network (CNN), which leverages the Edge Tensor Processing Unit (TPU) with the MobileNet SSD v2. The system was set up on a Jetson Nano with a few extras, including a camera and GPS, to support the IoT infrastructure. Evaluation for the model consists of device implementation, model evaluation, GPS position deviation, and on-road implementation. The effectiveness is confirmed through experiments using a system test-bed that generates ideal mAP off 0.22 and recall values.

Keywords— IoT; AI; Pothole Detection; CNN; MobileNet SSDV2.

I. INTRODUCTION

The Internet of Things (IoT) has been widely adopted worldwide because it can transparently and seamlessly integrate various disparate end systems [1]. With the innovation of IoT technology and the increased use of mobile devices, a brand-new computation paradigm, edge computing, is growing rapidly. Meanwhile, artificial intelligence (AI) applications are thriving due to deep learning breakthroughs and various hardware architecture improvements [2].

In addition to transportation, roads play a significant role in the community's market, societies, and cultures. The condition of the roads impacts several activities in the community. Road conditions will improve the quality of life in a given area and facilitate economic activity. Despite the efforts to improve, there are still areas of poor road quality, such as potholes. As reported by Indonesia's Ministry of Public Works and Housing, on the north coast road of Java Island, there are 3,338 potholes[3]. Roads too damaged for repairs must be marked with a traffic sign or signs, according to Indonesian Law Number 22/2009 No. 24 paragraph (2). This necessitates a damage survey and a public information system to display those results. Then, with the use of AI in cameras, it can help locate potholes. Hence, road damage detection has been explored extensively [5 - 23].

AI is one of the most widely used methods for detecting objects in cameras today [13]. A Deep Learning algorithm, Convolution Neural Network (CNN), is widely used in Machine Learning. In light of [14], using CNN in the introduction yields good results. When the CNN base and ResNet model are used, the pothole detection system improves its accuracy by 97.08 percent [15]. When analyzing images with the CNN (a deep neural network), its computational power is limited [16]. Since real-time surveying is portable, having a portable device with adequate processing power is important. Because of this, the Jetson Nano [17] and AI Accelerator [18][19] are used in combination. Systems-on-a-module (SoMs) such as Jetson Nano are popular computing modules for robotics and artificial intelligence use. In embedded systems, NVIDIA products, such as Jetson Nano, are frequently used for machine learning because of its small size and low power consumption [20]. Specifically, Google's own custom ASIC, the Edge TPU coprocessor [21], the AI accelerator speeds up the infringing process on Machine Learning models [22]. Google's AI accelerator developers have found success with and recommend using MobileNet SSD V2, combining MobileNet V2 and SSD (Single Shot Detector) to form a CNN model [23].

This paper proposes a systemic design and an implementation for detecting potholes with the Mobilenet SSD V2 model using CNN on Jetson Nano. The images of the potholes and their GPS coordinates will be sent to the server for analysis and display on the website. The system was installed on a test bed to perform performance evaluations using the Jetson Nano. It then conducted experiments to validate the proposal.

The rest of this article is structured as follows: Section II shows the study's related works and discusses the system's design. Section III evaluates the proposal through test-bed experiments. Finally, Section IV concludes this paper with future works.

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221

II. RESEARCH METHODOLOGY

A. Related Works

Here, the authors briefly touch on some other works relevant to this paper's topic. The research [4] presents multiple network models for the classification of road damage. They ascertained which algorithm works better for detecting and classifying road damage. The harm is categorized into potholes, cracks, and damage that exposes the road. This research employs R-CNN and faster R-CNN for road damage object detection and uses SVM for classification, and it achieves better results than previous research.

Authors present [14] develops a method for retrieving images of asphalt road holes based on their degree of damage by extracting features from the GLCM. This research aims to develop a retrieval system capable of detecting road damage based on its severity. Extraction of texture features from the GLCM. Specifically, 52 features were taken from 13 features at 0 degrees, 45 degrees, 90 degrees, and 135 degrees.

The research [18] discusses evaluating CNN has used several models with front-facing smartphone cameras to determine whether or not a road region contains potholes. They used a DCNN model, ResNet v2 152, ResNet v2, and MobileNet v1 are utilized to detect a pothole on the road surface. The study's results indicate that the best-performing models can accurately detect potholes in the road surface 96.5-97.5 percent of the time. Furthermore, they show that converting input frames to grayscale effectively boosts performance when detecting road potholes.

The research [19] proposes a CNN-based approach for classifying images as pothole/non-pothole that achieves better results than common SVM-based algorithms. A total of 13,244 images were used in the training process with a wide range of lighting, subject matter, and frame dimensions. The model performed exceptionally well when tested on new data, achieving an accuracy of 99.80%, a precision of 100%, a recall of 99.60%, and an F1-Score of 99.60%. However, results suffered greatly from varying illumination levels and significant gaps in the available data.

[20] emphasizes the importance of transfer learning. The CNN technique was demonstrated to identify and locate potholes in images. The proposed model is a spinoff of the YOLO model that reduces the neural network's computational costs and model size. From 48 million, the YOLO model's parameters are cut down to 18 million across 27 layers. Compared to the original YOLO architecture, the new modified design is smaller and runs faster while achieving higher average precision and recall scores.

In this review, the CNN model static picture was used by the majority of papers. However, no paper applies the model to real-world scenarios involving actively moving vehicles. As a result of the survey, we developed a system for detecting potholes by CNN on Jetson Nano to the MobileNet SSD V2 model, which should be able to locate and list road defects online.

B. Machine Learning Model Training

Our proposal is built on the TensorFlow Object Detection API [24]. In order to train networks on MobileNet SSD V2, this API was used.

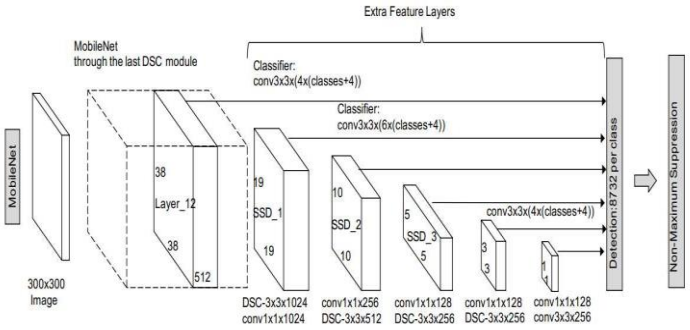


Figure 1. MobileNet SSD V2 Structure

The pre-trained SSD MobileNet V2 FPNLite 320x320' model is used. The data sets were formed from a combination of images taken from the internet, data test/training ratio of 70%:30%. Each TFRecord in the output directory was generated from the TFRecords in the input directory. The annotation information for each TFRecord file was included. Figure 1 displays the framework of a MobileNet SSD V2 [25]. In order to extract the feature from the input picture data, the network is split into two parts: combined efforts of a MobileNet v2 CNN model and a full-convolutional network using SSD for ROI proposal [35].

C. Addition of an AI Accelerator on the Jetson Nano

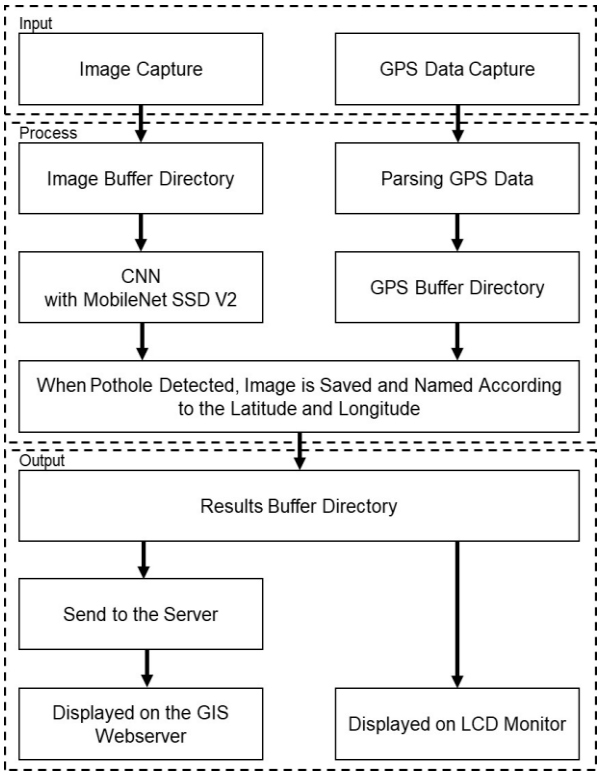


Figure 2. System Architecture

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221

Pothole detection is achieved in the post-processing step using the quantization technique, which employs a low-power, low-cost device for instantaneous monitoring. An integer approximation of floating-point computations used in neural networks for quantization during training, model accuracy can be restored to very close to the original by simulating the impact of quantization. Additionally, there is a size reduction of 4 times and an inference efficiency boost of 15% [36].

In order to use the AI accelerator, after the quantizing model was complete, further processing was necessary. The compilation is necessary to pass model-specific operations to the USB accelerator; the central processing unit handles everything else. In this study, the TensorFlow Object Detection API was used for inference. The File Uploader and CNN inference process both listen to the temporary folder and upload the picture to the server over the internet. As illustrated in Figure 2, the system uses this architecture process.

D. Device Implementation

Adding an AI accelerator to the Jetson Nano allows the construction of a system with a compact design and ease of processing while maintaining the required processing power.

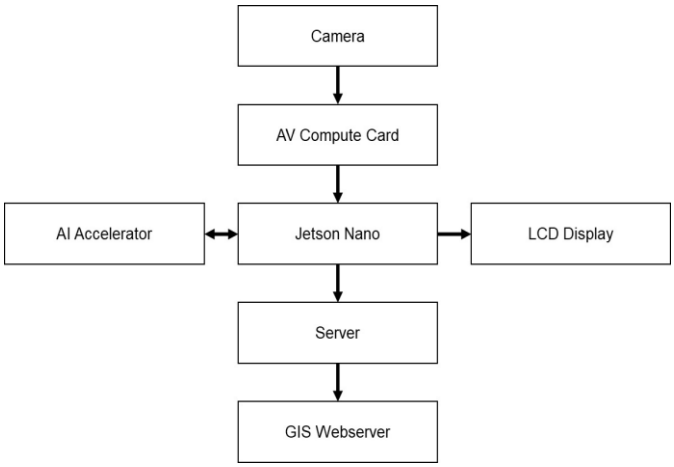


Figure 3. System Integration Architecture

Figure 3 illustrates the integration architecture for each component. Input, processing, connectivity, and output are all system components. Due to the quality of the source material—a rear-mounted car camera—an AV video capture device is required to transform the raw footage into a format that the Jetson Nano can read. The device employs a small, modified USB connector to reduce the overall size. The Jetson Nano is integrated into the AI accelerator for its processing. A mini-display unit, an audio-visual capture card, a wireless LAN dongle, and some other peripherals were all wired straight into the controller USB port.

The 3D-printed case provides structural integrity and makes the implemented device more user-friendly. A battery, controller, and other required parts will round out the finished product. The necessary on/off button, switches controls, charging, and RCA ports are all set up. Also included is a mini digital voltmeter for checking battery life.

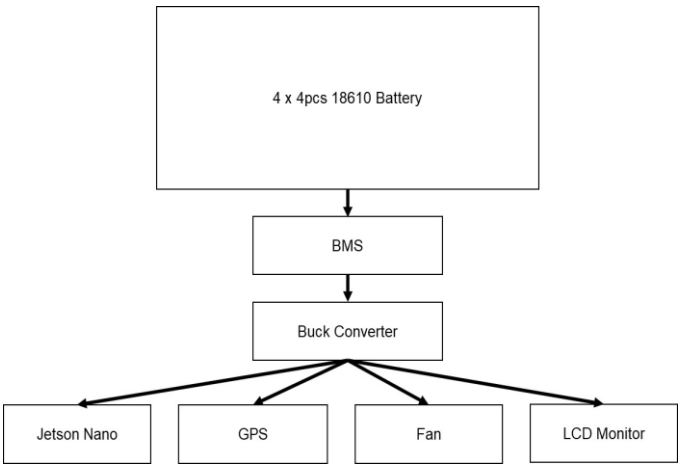


Figure 4. Electrical Block Diagram

Figure 4 illustrates the electrical block diagram used in this implementation.

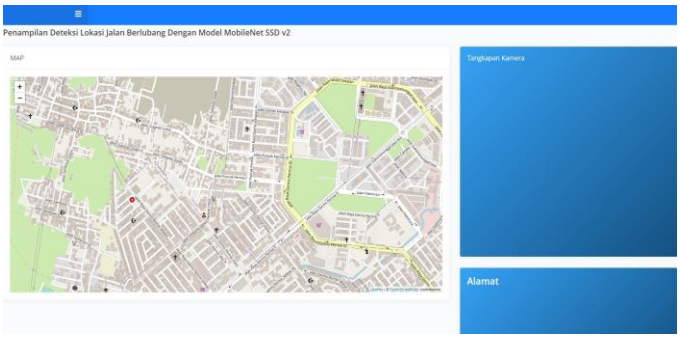


Figure 5. Geographic Information System

TABLE I
DEVICE SPECIFICATIONS

Main Controller	
Model	Model Jetson Nano
GPU	NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores
CPU	Quad-core ARM Cortex-A57 MPCore processor
Memory	4 GB 64-bit LPDDR4, 1600MHz 25.6 GB/s
Operating System	Linux
Peripheral	
AI Accelerator	Google Coral
Camera	AV Car Camera 170degree
WIFI	Realtek RTL8188EUS
LoRa	SEMTECH SX1276
GPS	Neo M9N

The photos of the potholes are shown on a dynamic website. The geographic information system appearance plots the marker concerning the actual pothole location. In this paper, an interactive map was developed using the free and open-source Leaflet JavaScript library. Images are uploaded to

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221

a server using the PHP POST method, with coordinates incorporated into the filename. Figure 5 displays a website that was set up. Table I reveals the device and software specifications used in the experiments. One Jetson Nano is employed as the main controller.

III. Result and Discussion

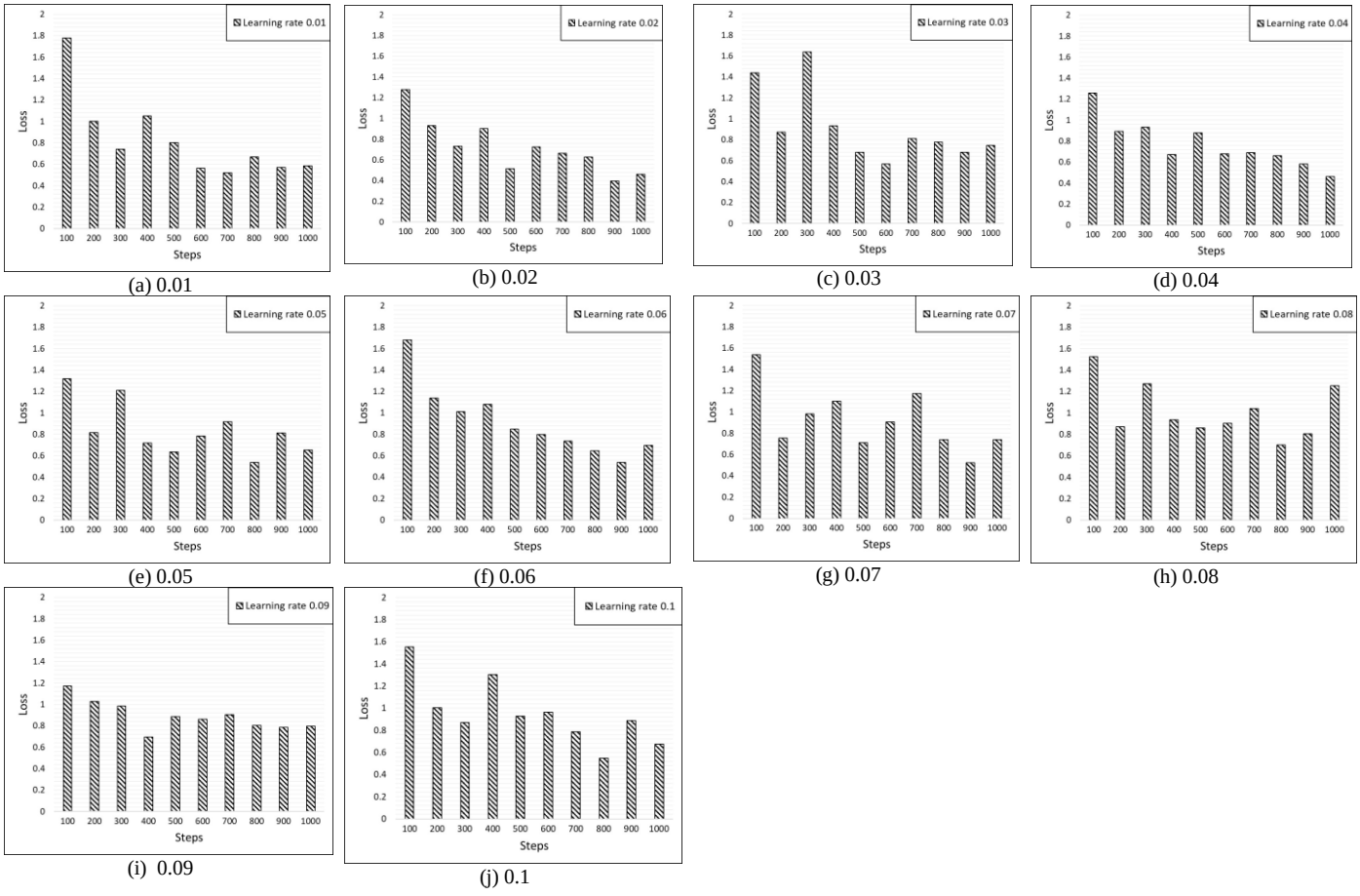
A. Device Implementation

This pre-trained model was used to train the Predictor, and the optimal combination of hyper-parameters was obtained by experimenting with different learning rates. A different learning rate was used to train the model, and only a thousand iterations of transfer learning were used to calculate the corresponding loss value. The training parameter is shown in Table II, where the pothole image is collected on the national road between Surabaya City – Banyuwangi City – Jember City.

TABLE II
HYPER PARAMETER TO TRAIN CNN

Total Images	1700 Pothole images
Data Test	30%
Data Training	70%
Batch Size	62

As illustrated in Figure 6, both 0.01 and 0.10 make for excellent learning rates. Seventy-five thousand iterations of Single-Predictor transfer learning at a learning rate of 0.10. A 64-batch maximum was used in the gradient descent update, one of the procedures. The total loss of a CNN model generated using the COCO Evaluation Metric method is 0.1, and the mAP is 0.22.



(a) 0.01 (b) 0.02 (c) 0.03 (d) 0.04
(e) 0.05 (f) 0.06 (g) 0.07 (h) 0.08
(i) 0.09 (j) 0.1

Figure 6. Per-step Lost with Various Learning Rate Values

B. Evaluation of Model Training

The inference procedure was executed on each device following the acquisition of videos of a car driving at various speeds. Table III summarizes the results of these experiments.

From this data, the total of FP is clearly relatively high, and night-time result performs extremely poorly, but the detection process itself is quite fast. Figure 7 demonstrates the pothole detection result based on the tested data. The figure also shows how road roughness and engine vibration cause the tested image to blur at high speeds. Likewise, the testing process demonstrates that the faster the vehicle detects, the fewer correct results it generates.

TABLE III
DETECTION IN DIFFERENT SPEED

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221

Speed (Km/h) 0-10	TP (frame) 487	FP (frame) 483	Total (frame) 970
10-20	258	299	557
20-30	261	387	648
30-40	110	529	639
Night	6	173	179

(a) Speed 0-10 km/h

(b) Speed 10-20 km/h

(c) Speed 20-30 km/h

(d) Speed 30-40 km/h

Figure 7. Conditions on A Different Speed

C. GPS Position Deviation

To calculate the GPS error, compare the observed and actual locations. Vincenty's formulas [28] determine the distance between two latitudes and longitudes. We gathered the data from various pothole spots and sent to the server. The result is shown in Table IV; a standard deviation of 1.5 meters exists between the actual location of the pothole and the location due to the evaluation process.

TABLE IV
GPS DEVIATION

Actual Lat., Lon.	Reported Lat., Lon.	Deviation (m)
-7.279,112.789	-7.266,112.783	1,5
-7.253,112.795	-7.250,112.783	1,3
-7.255,112.795	-7.250,112.783	1,4
-7.271,112.797	-7.266,112.783	1,5
-7.273,112.797	-7.266,112.783	1,7
-7.274,112.797	-7.266,112.783	1,7
-7.279,112.772	-7.266,112.783	1,8
-7.278,112.772	-7.266,112.783	1,7

D. Deployment in the Field

Device configuration in the situation represented. See the evaluations in Figures 8 and 9 for how the device is actually used. The camera is mounted in the vehicle's front end and controlled from within. Surabaya, Indonesia, was chosen as the location for the survey.

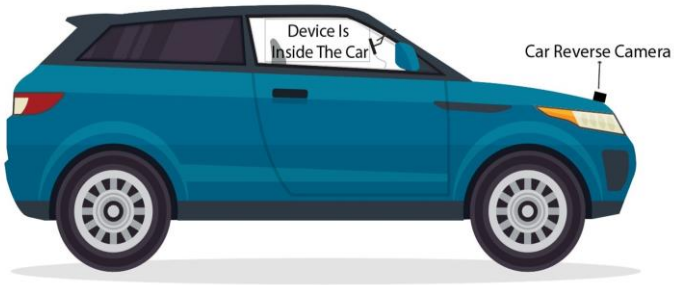


Figure 8. Device Implementation on Vehicle

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221

Figure 9. Camera Implementation on Vehicle.

E. Website Capacity Evaluation

The JMeter [29] tool is used for website capacity evaluation in this evaluation. This JMeter tool can simulate the number of users who make requests to the server at the same time. This JMeter software tool will make requests with a variable number of users within one second, allowing the website's traffic performance to be determined. Table V shows the JMeter test results of the website's traffic capacity.

TABEL V
WEBSITE TRAFFIC EVALUATION USING JMETER

Number of Users	Average Response Time	Average Latency	Error %
100	239	164	0
300	162	88	0
600	469	237	0
900	389	185	0
1200	666	234	0
1500	1220	747	0

These evaluations show that as the number of users accessing the website increases, the response time and server latency also increase. However, the time begins to be felt when the number of users exceeds 1000, and no errors are detected in any requests made.

IV. CONCLUSION

A CNN method was used to effectively finish the design and construction of an AI-based real-time pothole detecting system for IoT smart infrastructure. This proposal includes an experiment that shows transmission and real-time presentation of image data depicting potholes. The Convolutional Neural Network model used in this paper has a TP rate of about 25% even though it runs at 60 frames per second on the device. Furthermore, our implemented GPS has a deviation of 1.5 meters. That indicates that our experiment was successful. In this paper, we implemented real-time processing and extensively tested the inference procedure. In the future, we can optimize data acquisition to reduce server latency. Put forth a fresh approach to pothole warnings and adaptive suspension for future cars. This tool and a lidar array can determine how rough a road surface is. A more cohesive outcome calls for both improved datasets and updates to the CNN model are being made.

ACKNOWLEDGMENT

The authors would like to thank Politeknik Elektronika Negeri Surabaya, PT. Telekomunikasi Indonesia and the Ministry of Education, Culture, Research, and Technology for the support in the realization of the EMSiT (EEPIS Mobile Smart City Based on IoT). The authors would also like to express their appreciation to the engineers who helped with the system's deployment and supplied the experimental data and technical documentation for the EMSiT project.

REFERENCES

[1] B. B. V. Rao, H. G. Chandrakanth, "IoT based Machine Learning Automation Algorithm for Controlling the Industrial Loads", in *Int. J. of Intelligent Engineering and Systems*, Vol.14, No.4, pp. 146-156, Aug. 2021.

[2] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence", in *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457-7469, Aug. 2020.

[3] I. R. Putra, "Ministry of Public Works: 3.338 Holes along the North Coast Road in Java Island", https://www.merdeka.com/uang/kemen-terian-pu-ada-3338-lubang-di-sepanjang-jalur-pantura.html. Accessed 18 Nov. 2022.

[4] Md. S. Arman, Md. M. Hasan, F. Sadia, A. K. Shakir, K. Sarker, and F. A. Himu, "Detection and Classification of Road Damage Using RCNN and Faster R-CNN: A Deep Learning Approach", in Cyber Security and Computer Science. ICONCS 2020. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 325. Springer, July 2020.

[5] R. Mukherjee, H. Iqbal, S. Marzban, A. Badar, T. Brouns, S. Gowda, E. Arani, and B. Zonoos, "AI Driven Road Maintenance Inspection," Proc. 27th ITS World Congress, June 2021.

[6] S. Naddaf-Sh, M. M. Naddaf-Sh, A. R. Kashani, and H. Zargarzadeh, "An Efficient and Scalable Deep Learning Approach for Road Damage Detection," Proc. 2020 IEEE Int. Conf. Big Data, pp. 5602–5608, 2020.

[7] Z. Liu, W. Wu, X. Gu, S. Li, L. Wang, and T. Zhang, "Application of combining yolo models and 3d gpr images in road detection and maintenance," Int. J. Remote Sens., vol. 13, no. 6, pp. 1–18, 2021.

[8] Z. Pei, R. Lin, X. Zhang, H. Shen, J. Tang, and Y. Yang, "CFM: A Consistency Filtering Mechanism for Road Damage Detection," Proc. 2020 IEEE Int. Conf. Big Data, pp. 5584-5591, 2020.

[9] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," Int. J. Autom. Constr., vol. 132, 2021.

[10] Y. Liu, X. Zhang, B. Zhang, and Z. Chen, "Deep Network for Road Damage Detection," Proc. 2020 IEEE Int. Conf. Big Data, pp. 5572–5576, 2020.

[11] F. Kortmann, K. Talits, P. Fassmeyer, A. Warnecke, N. Meier, J. Heger, P. Drews, and B. Funk, "Detecting Various Road Damage Types in Global Countries Utilizing Faster R-CNN," Proc. 2020 IEEE Int. Conf. Big Data, pp. 5563–5571, 2020.

[12] T. Hascoet, Y. Zhang, A. Persch, R. Takashima, T. Takiguchi, and Y. Ariki, "Faster RCNN Monitoring of Road Damages: Competition and Deployment," Proc. 2020 IEEE Int. Conf. Big Data, pp. 5545–5552, 2020.

[13] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2020: An annotated image dataset for automatic road damage detection using deep learning," Int. J. Data in Brief, vol. 36, 2021.

[14] A. Mahardika, Y. Sari, and C. Dewi, "Sistem Temu Kembali Citra Lubang Jalan Aspal Berdasarkan Tingkat Kerusakan Menggunakan

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221

Ekstraksi Fitur Gray Level Co-occurrence Matrix", in *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, pp. 3811-3821, Feb. 2018.

[15] H. Maeda, Y. Sekimoto, and T. Seto, "Lightweight Road Manager: Smartphone-based Automatic Determination of Road Damage Status by Deep Neural Network", in *MobiGIS '16: Proc. of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pp. 37–45, 2016.

[16] I. D. Pratama, H. Mahmudah, and R. W. Sudibyo, "Design and Implementation of Real-time Pothole Detection using Convolutional Neural Network for IoT Smart Environment", in *Proc. 2021 Int. Electron. Symp. IES pp. 675–679, 2021*.

[17] Z. S. Hernanda, H. Mahmudah, and R. W. Sudibyo, "CNN-Based Hyperparameter Optimization Approach for Road Pothole and Crack Detection Systems", in *Proc. 2022 World AI IoT Congress pp. 549–554, 2022*.

[18] K. E. An, S. W. Lee, S.-K. Ryu, and D. Seo, "Detecting a Pothole using Deep Convolutional Neural Network Models for an Adaptive Shock Oserving in a Vehicle Driving", in *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-2, 2018.

[19] V. Pereira, S. Tamura, S. Hayamizu, and H. Fukai, "A deep learning based approach for road pothole detection in Timor Leste", in *Proc. IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 279-284, 2018.

[20] L. K. Suong and J. Kwon, "Detection of Potholes Using a Deep Convolutional Neural Network", in *Int. J. UCS*, vol. 24, no. 9, pp. 1244– 1257, 2018.

[21] Y. Y. F. Panduman, A. R. A. Besari, S. Sukaridhoto, R. P. N. Budiarti, R. W. Sudibyo, and F. Nobuo, "Implementation of Integration VaaMSN and SEMAR for Wide Coverage Air Quality Monitoring", in *Int. Journal TELKOMNIKA*, vol. 16, no. 6, p. 2630, Dec. 2018.

[22] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection with Deep Learning: A Review", in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, Nov. 2019.

[23] W. S. E. Putra, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101", in *Jurnal Teknik ITS*, vol. 5, no. 1, pp. A65-A69, 2016.

[24] Aparna, Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional Neural Networks based Potholes Detection using Thermal Imaging", in *Journal of King Saud University - Computer and Information Sciences*, Feb. 2019.

[25] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The Computational Limits of Deep Learning", in *arXiv:2007.05558v1*, July 2020.

[26] NVIDIA, *Jetson Nano Developer Kit*, https://developer.nvidia.com/embedded/jetson-nano-developer-kit. Accessed 18 Nov. 2022.

[27] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey of Machine Learning Accelerators", *in arXiv:2009.00993v1*, Sep. 2020.

[28] W. Li and M. Liewig, "A Survey of AI Accelerators for Edge Environment", in Trends and Innovations in Information Systems and Technologies. WorldCIST 2020. Advances in Intelligent Systems and Computing, vol 1160. Springer, Cham, Jun. 2020.

[29] A. Basulto-Lantsova, J. A. Padilla-Medina, F. J. Perez-Pinal, and A. I. Barranco-Gutierrez, "Performance Comparative of OpenCV Template Matching Method on Jetson TX2 and Jetson Nano Developer Kits", in *Proc. 10th Annual Computing and Communication Workshop and Conference (CCWC)*. pp. 0812-0816, Jan. 2020.

[30] Google Coral, *Build Beneficial and Privacy Preserving AI*, https: //coral.ai/. Accessed 18 Nov. 2022.

[31] L. A. Libutti, F. D. Igua, L. Pi˜nuel, and L. D. Giusti, "Benchmarking Performance and Power of USB Accelerators for Inference with MLPerf", in *2nd Workshop on Accelerated Machine Learning (AccML)*, Spain 2020.

[32] Coral Benchmark, *Edge TPU Performance Benchmarks*, https://coral.ai/docs/edgetpu/-benchmarks/. Accessed 18 Nov. 2022.

[33] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al, "Speed/accuracy trade-offs for modern convolutional object detectors", in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 7310–7311, 2017.

[34] Y. Zhang, H. Pheng, and P. Hu, "CS341 Final Report: Towards Real-time Detection and Camera Triggering", in 2017.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[36] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference", in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[37] T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equation", in *Survey Review Ministry of Overseas Development Surrey*, vol. XXIII. no. 176, Apr. 1975.

[38] Apache JMeter, https://jmeter.apache.org/. Accessed 18 Nov. 2022.

DOI: http://dx.doi.org/10.25139/inform.v8i2.6221