



Edge Computing Implementation for Action Recognition Systems

Afis Asryullah Pratama¹, Yohanes Yohanie Fridelin Panduman², Dwi Kurnia Basuki³ and Sritrusta Sukaridhoto⁴

^{1,3,4}Informatics and Computer Departement, Electronics Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia

²Electrical Engineering Departement, Electronics Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia

Email: ¹afisarsy@gmail.com, ²panduyohan8@gmail.com, ³dwiki@pens.ac.id, ⁴dhoto@pens.ac.id

Abstract

Nowadays the deep learning has been improved to many different sectors, including human action recognition system. This system mostly needs a high computing resource to work on. Its implementation will be built under cloud computing architecture, which requires sensors used to send whole raw data to the cloud, which puts a load in the networks. Therefore, edge computing system exists to overcome that weakness. This paper presents a method to recognize human action using deep learning with edge computing architecture. With RGB image as the input, this system will detect all persons in the frame using SSD-MobileNet V2 model with various threshold values, then recognize every person's action using our trained model with DetectNet architecture in various thresholds too. The system's output is detected person's RoI and its recognized action, which a lot smaller than the whole frame. As a result, our proposed system yields the best accuracy of human detection at 64.06% with a threshold at 0.15 and the best accuracy of action recognition at 37.8% with a threshold at 0.4.

Keywords: Computer vision, Deep learning, IoT, Edge Computing

1. INTRODUCTION

IoT with Deep learning development is still expanding through many different sectors, including human action recognition system. For the implementation, this system needs some readjustment due to its high computing resource and proper network needs. The development of devices that support the internet of things in the industrial sector is a device for collecting data and combined with an automation system into a technological concept known as Smart Industry [1]. However, the current trend of the manufacturing revolution is leading to the integration of several industrial technologies, thus giving birth to Industry 4.0 [2]. The application of emerging technologies in industry 4.0 has changed manufacturing activities such as the production process, such as how products are produced, how products are marketed, how products or raw materials are delivered, and how workers carry out their activities [3]. The challenge in developing the

latest 4.0 industrial architecture is not only centered on manufacturing technology but also on humans as workers to ensure their conditions to optimize the production process [4]. There are many methods for monitoring workers, such as using wearable devices such as BLE [4] and accelerator [5] or using cameras [6] to detect worker gestures and movements combined with artificial intelligence (AI). However, the application of AI is carried out on the server so that the detection process takes a long time and requires greater resources. Thus, in this research focused on a solution for implementing edge computing technology for the detection process using AI. Edge computing is a new technological paradigm in which computing and storage processes are located close to mobile devices or sensors [7], thus, that data processing on the server is not too heavy. One of the ongoing research on the implementation of edge computing technology used for detection and monitoring processes [8]. The application of Edge computing technology uses the Vehicle as a Mobile Sensor Network (VaaMSN) device to receive and process air quality sensor data that is placed on a vehicle [8]. Edge computing technology can be combined with IMU sensors to detect road holes based on accelerometer and gyro values [9]. This research has been enhanced by adding a camera sensor to edge computing to detect road changes based on continuous camera-generated images, the pothole detection process is carried out on edge computing devices [10]. Therefore, in this study, edge computing will be implemented which implements a worker action recognition system combined with a person detection system, so that it can monitor the performance and activities of workers, the resulting data can be used to improve the performance and efficiency of the manufacturing process.

Therefore, in this research, an edge computing action recognition system that gives an output of processed data rather than raw data to suppress network load was proposed.

2. METHODS

Our proposed system combines image-based action recognition with edge computing, this research uses a camera and NVIDIA Jetson Nano as a computing system; therefore, it could give a wider way of implementation due to its speed, less communication load, and high mobility. The overall system design is shown in Figure 1.

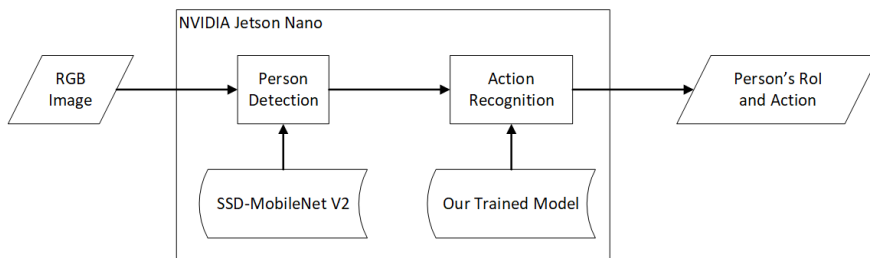


Figure 1. System design

2.1. Obtain the Image

In this research, an ESCAM QD900 WIFI IP Camera is used to get the visual image of person's action with a 2MP effective pixel. The camera output was set to RGB 1920x1080 pixel and connected to the same network with NVIDIA Jetson Nano to provide data transfer through RTSP.

2.2. Computing System

NVIDIA Jetson Nano was used as an edge computing device where all of the deep learning systems are implemented. NVIDIA Maxwell GPU and Jetson Inference Library could give a high performance of continuous deep learning process. Two kinds of deep learning were used: Detectnet [11], which is used as person detection and Imagenet [12] for action recognition, both provided in the Jetson Inference library.

2.3. Person Detection

Person detection use Detectnet architecture from Jetson Inference with a pre-trained SSD-MobileNet V2 model provided by NVIDIA. SSD-MobileNet V2 is a high-accuracy object detection model with various categories of objects, including the person category; therefore, a person label filter was needed. The inference process was tested with threshold values at 0.1, 0.15, 0.2, 0.25, 0.3, and 0.4 to find the best threshold of person detection at a distance around 3m away from the camera. The person detection process would give an output of each detected person's ROI which will be cropped. Each cropped person image will be used as an input of action recognition. Our person detection process is shown in Figure 2.

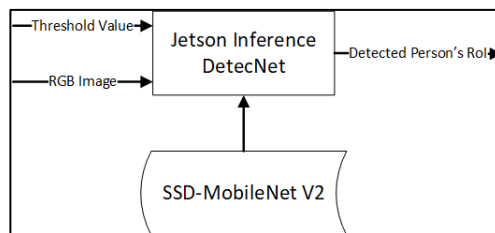








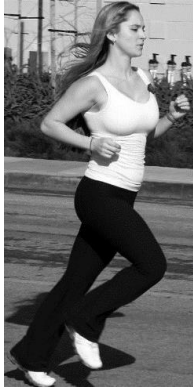





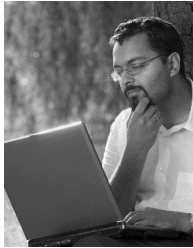


Figure 2. Person detection process

2.4. Action Recognition

Dataset was built by combining the N-UCLA dataset [13], UCF Sports dataset [14], Stanford 40 Actions dataset [15], Willow Actions dataset [16], and INRIA Person Dataset [17] and cropped it for each person in the images to provide 6371 person images with seven action categories, which are: carrying things, pick up, sit down, running, using a computer, writing, and walking as in Table 1.

Table 1. Datasets

Carrying things			
Pick up			

Running			
Sit down			
Using a computer			



NVIDIA DIGITS were used to train our dataset with following dataset configuration as in Figure 3.

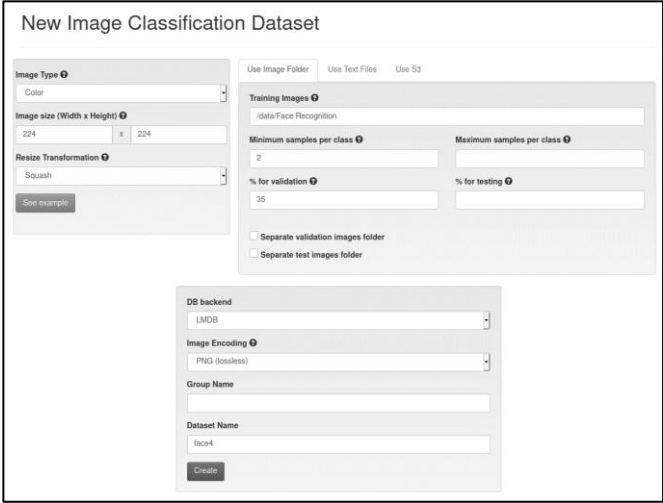


Figure 3. NVIDIA DIGITS Dataset configuration

NVIDIA DIGITS is a training tool that supports DetectNet and ImageNet architectures. NVIDIA DIGITS Trained model compatibility with Jetson Inference was our consideration of using it. The configuration used to train the dataset is shown in Figure 4.

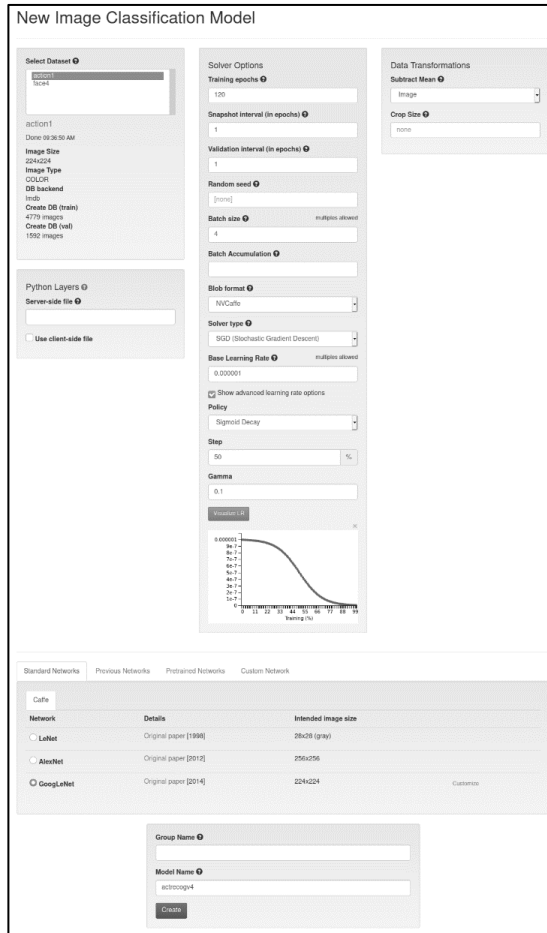


Figure 4. NVIDIA DIGITS Training configuration

Our action recognition process is shown in Figure 5. The system use Imagenet from Jetson Inference with our action recognition model.

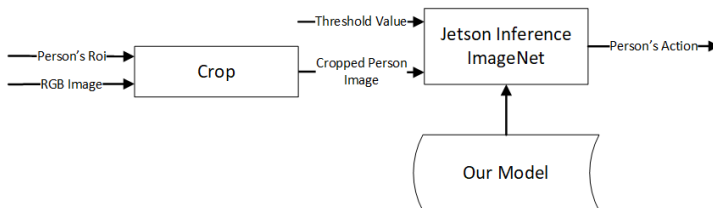


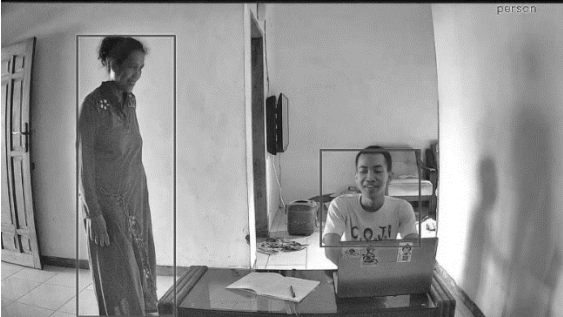


Figure 5. Action recognition process

The inference threshold value was set to 0.2, 0.3, 0.4, 0.5, 0.6, and 0.7 to find the best threshold for our action recognition.

3. RESULT AND DISCUSSION

The system was tested by performing actions at distance about 3 meters in front of the camera and provide the result of person detection and action recognition test. The output of person detection test was divided into the following categories as in Table 2.

Table 2. Person detection output categories

Result	Categories
	Correct
	False negative
	False positive



Multi-detect



Not detected



Partial
detect

Based on the six output categories, the accuracy of correct detection from each threshold value is shown in Table 3.


Table 3. Accuracy of person detection with a various threshold value

Threshold	Correct	Total Images	Accuracy (%)
0.1	69	128	53.91
0.15	82	128	64.06
0.2	81	128	63.28
0.25	79	128	61.72
0.3	78	128	60.94
0.4	71	128	55.47

Thus, the best accuracy for person detection at 64.06% with a threshold value at 0.15.

For action recognition, the output was divided into three categories as shown in Table 4.

Table 4. Action recognition output categories

Result	Categories
	Correct



Wrong



Unknown

Using person detection with the threshold value at 0.15, the accuracy of correct action recognition for each threshold value is shown in Table 5.

Table 5. Accuracy of action recognition at a various threshold value

Threshold	Correct	Total Images	Accuracy (%)
0.2	31	82	37.8
0.3	31	82	37.8
0.4	31	82	37.8
0.5	30	82	36.59
0.6	29	82	35.37
0.7	27	82	32.93

From the experiment using person detection with the threshold value at 0.15, the best accuracy saturated at 37.8% starts from threshold value at 0.4.

4. CONCLUSION

Based on experiments conducted, it can be concluded that the action recognition system could be integrated with edge computing architecture to reduce the network load. Our proposed system could detect person with the best accuracy at 64.06% with the threshold value at 0.15 and the best accuracy of action recognition at 37.8 with a threshold value at 0.4.

5. REFERENCES

- [1] Zhuming Bi, Li Da Xu, and Chengen Wang, "Internet of Things for Enterprise Systems of Modern Manufacturing," *IEEE Trans. Ind. Informatics*, 10(2), pp. 1537–1546, May 2014, doi: 10.1109/TII.2014.2300338.
- [2] J. Wan *et al.*, "A Manufacturing Big Data Solution for Active Preventive Maintenance," *IEEE Trans. Ind. Informatics*, 13(4), pp. 2039–2047, Aug. 2017, doi: 10.1109/TII.2017.2670505.
- [3] A. G. Frank, L. S. Dalenogare, and N. F. Ayala, "Industry 4.0 technologies: Implementation patterns in manufacturing companies," *Int. J. Prod. Econ.*, vol. 210, pp. 15–26, Apr. 2019, doi: 10.1016/j.ijpe.2019.01.004.
- [4] L. Roda-Sanchez, C. Garrido-Hidalgo, D. Hortelano, T. Olivares, and M. C. Ruiz, "OperABLE: An IoT-Based Wearable to Improve Efficiency and Smart Worker Care Services in Industry 4.0," *J. Sensors*, vol. 2018, pp. 1–12, Aug. 2018, doi: 10.1155/2018/6272793.
- [5] M. Nguyen, L. Fan, and C. Shahabi, "Activity Recognition Using Wrist-Worn Sensors for Human Performance Evaluation," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 164–169, doi: 10.1109/ICDMW.2015.199.
- [6] M. Neuhausen, J. Teizer, and M. König, "Construction worker detection and tracking in bird's-eye view camera images," *ISARC 2018 - 35th Int. Symp. Autom. Robot. Constr. Int. AEC/FM Hackathon Futur. Build. Things*, no. Isarc, 2018, doi: 10.22260/isarc2018/0161.
- [7] M. Satyanarayanan, "The Emergence of Edge Computing," *Computer (Long. Beach. Calif.)*, vol. 50, no. 1, pp. 30–39, Jan. 2017, doi: 10.1109/MC.2017.9.
- [8] 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," *Telkomnika (Telecommunication Comput. Electron. Control.)*, vol. 16, no. 6, pp. 2630–2642, 2018, doi: 10.12928/TELKOMNIKA.v16i6.10152.
- [9] A. Mochamad Rifki Ulil, Fiannurdin, S. Sukaridhoto, A. Tjahjono, and D. K. Basuki, "The Vehicle as a Mobile Sensor Network base IoT and Big Data for Pothole Detection Caused by Flood Disaster," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 239, p. 012034, Feb. 2019, doi: 10.1088/1755-1315/239/1/012034.
- [10] A. Rasyid *et al.*, "Pothole Visual Detection using Machine Learning Method integrated with Internet of Thing Video Streaming Platform," *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 672–675, 2019, doi:

- 10.1109/ELECSYM.2019.8901626.
- [11] Andrew Tao, Jon Barker, and S. Sarathy, “DetectNet: Deep Neural Network for Object Detection in DIGITS | NVIDIA Developer Blog,” 2016. <https://developer.nvidia.com/blog/detectnet-deep-neural-network-object-detection-digits/> (accessed Sep. 30, 2020).
 - [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, 2017, doi: 10.1145/3065386.
 - [13] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. C. Zhu, “Cross-view action modeling, learning, and recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2649–2656, 2014, doi: 10.1109/CVPR.2014.339.
 - [14] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action MACH: A spatio-temporal maximum average correlation height filter for action recognition,” *26th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*, 2008, doi: 10.1109/CVPR.2008.4587727.
 - [15] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, “Human action recognition by learning bases of action attributes and parts,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1331–1338, 2011, doi: 10.1109/ICCV.2011.6126386.
 - [16] V. Delaitre, I. Laptev, and J. Sivic, “Recognizing human actions in still images: A study of bag-of-features and part-based representations,” *Br. Mach. Vis. Conf. BMVC 2010 - Proc.*, 2010, doi: 10.5244/C.24.97.
 - [17] N. Dalal *et al.*, “Histograms of Oriented Gradients for Human Detection To cite this version : HAL Id : inria-00548512 Histograms of Oriented Gradients for Human Detection,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 886–893, 2010.