Effects of Video Filters for Learning an Action Recognition Model for Construction Machinery from Simulated Training Data

Jinhyeok Sim[1], Jun Younes Louhi Kasahara[1], Shota Chikushi[1], Keiji Nagatani[1], Takumi Chiba[2], Kazuhiro Chayama[2], Atsushi Yamashita[1],and Hajime Asama[1]

***Abstract*—In the construction industry, construction machinery are an important factor in the overall productivity and efficiency of a worksite. Thus, emphasis is put on the monitoring of actions conducted by construction machinery. This was traditionally done manually by humans, which is a time-consuming and laborious task. Automatic action recognition of construction machinery is therefore needed. The field of action recognition is predominantly occupied by Deep Learning approaches and several previous works focused on adapting such approaches for construction machinery. However, the issue of obtaining training data is particularly troublesome for construction machinery. Our previous work proposed a Deep Learning method for learning an action recognition model from training data generated in a simulator using video filters but the precise contributions of the introduced video filter were unclear. The purpose of this study is therefore to clarify the effects of video filters for learning an action recognition model for construction machinery from simulated training data.**

I. Introduction

The construction industry is known to suffer from poor efficiency compared to other industries such as manufacturing [1]. One key factor in improving the efficiency at construction sites is the monitoring of construction machinery, which occupy large portions of the overall budget [2]. Indeed, precise knowledge of the time and costs associated with each action conducted by each specific construction machinery is paramount in the establishment of an efficient construction plan [3].

Action recognition of construction machinery was traditionally conducted manually by site workers [4]. However, the need for automation have motivated several previous works. Those can be distinguished between approaches using onboard sensors and those using outboard sensors. Onboard sensors usually consist of encoders or GPS [5] and can provide reliable data for action recognition. However, they involve modifications on existing construction machinery. Outboard sensors consist of in cameras [6] or microphones [1] positioned throughout the construction site. Those allow direct use of preexisting construction machinery as well as monitoring of several construction machinery with a single sensor. Approaches using cameras have especially boomed, partly due to the success of Deep Learning-based approaches to human action recognition [7][8], and several Deep Learning-based approaches for action recognition of construction machinery using cameras were proposed [9][10]. However, Deep Learning approaches require training data, which is troublesome to obtain for construction machinery. Our previous work [11] proposed a novel approach for learning a model from simulator-generated training data, much easier to obtain than training data from actual construction sites. The

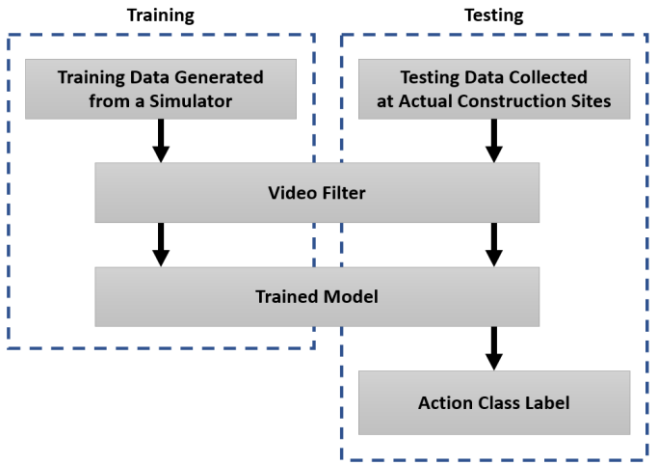

Figure 1. Overview of proposed method.

Figure 2. Simulator environment used in our proposed method.

[1] Jinhyeok Sim, Jun Younes Louhi Kasahara, Shota Chikushi, Keiji Nagatani, Atsushi Yamashita and Hajime Asama are with the Department of Precision Engineering, The University of Tokyo, 113-8656 Tokyo, Japan. (e-mail: sim@robot.t.u-tokyo.ac.jp).

[2] Takumi Chiba and Kazuhiro Chayama are with Fujita Corporation, 151-0051, Tokyo, Japan.

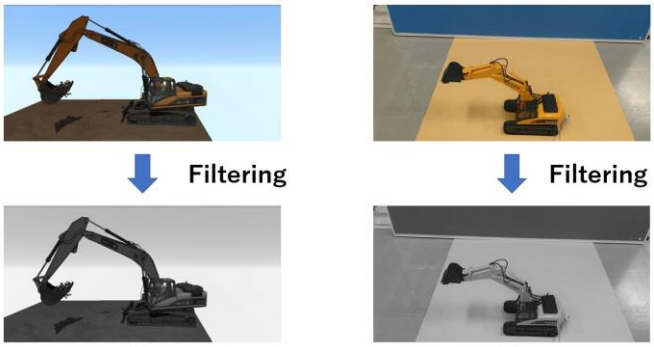

Figure 3. Effects of applying a Grayscale filter: the color differences between (a) the data generated from a simulator and (b) from the real world have their color differences suppressed.

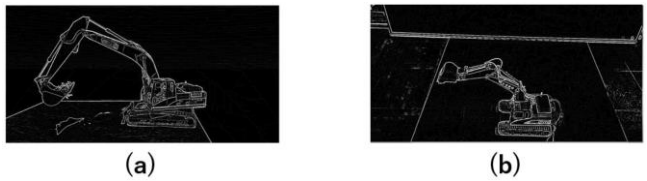

Figure 4. Effects of applying an Edge filter: both (a) the data generated from a simulator and (b) from the real world have their texture differences suppressed.

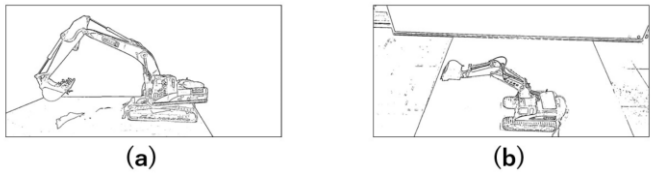

Figure 5. Effects of applying our filter: both (a) the data generated from a simulator and (b) from the real world have unnecessary edges removed.

learning problem was shifted to a common feature space domain by using video filters. However, the exact effects of video filters on the learning process were left unclear. Therefore, in this paper, the effects of video filters for learning an action recognition model for construction machinery from simulated training data are clarified.

II. Action Recognition of Construction Machinery based on Simulated Training Data

A. Concept

An action recognition model trained using training data generated in a simulator performs poorly on real world data due to the mismatch in feature space domain. Indeed, such training data would not be appropriate: the features contained in data generated in a simulator are not the same as those contained in real world data. This is most obvious by looking at them: the construction machinery in the simulator does not look similar to the one in real construction sites.

In order to solve the mismatch in feature space domain, a third domain is introduced using video filters. By converting both the simulator generated training data and the real world data into the feature domain arbitrary defined by the selected filter, the learning problem can be contained in a feature domain where both data are represented.

An overview of the method is shown in Fig. 1. Training is conducted using training data generated from a simulator after application of a video filter. Prior to testing, this same video filter is applied to the data collected at actual construction sites.

B. Simulator-Generated Training Data

A large amount of training data can be easily generated using a simulator. In this study, Vortex Studio [12], a real- time simulator for mechanical system operation, was used with a model of an excavator, as illustrated in Fig. 2.

The input data to our learning model is RGB video data and to limit the effects of background, an environment with only soil around the excavator was created. Multiple camera viewpoints were also considered in the training data generation. During recording, the movements of the excavator were controlled by a human using a controller.

C. Video Filters

The videos segments generated from the simulator differ from real world video segments in several ways. First is about colors. Construction machinery and the background at construction sites come in a multitude of colors and are likely to differ from the selected ones in the simulator. Therefore, the first step consists in removing colors from video segments. This is simply done by the use of a Grayscale filter, shown in Fig. 3, which returns an image with pixel intensity values $I(x, y)$, with (x, y) being pixel coordinates.

The second disparity is regarding textures. Surfaces in real world are not uniform, e.g., on construction machinery colors vary greatly in hue and contrast due to wear. Construction machinery in a simulator by opposition have unnaturally uniform surfaces. To suppress this difference, an Edge filter, shown in Fig. 4, is applied to preserve the contours of objects and remove any information regarding textures. This Edge filter, consisting of a Laplacian filter as in (1), is applied following the Grayscale filter.

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}. \quad (1)$$

However, the Edge filter can be expected to generate not only contours of objects necessary for action recognition but also unnecessary edges. Those additional edges would hinder the recognition task, creating additional differences between the simulator generated training data and the real world test data, and are therefore removed. This is conducted by thresholding with parameter T as in (2) to obtain the final output of our filter $L^*(x, y)$, shown in Fig. 5.

$$L^*(x, y) = \begin{cases} 0, & \text{if } L(x, y) \geq T \\ 255, & \text{otherwise.} \end{cases} \quad (2)$$

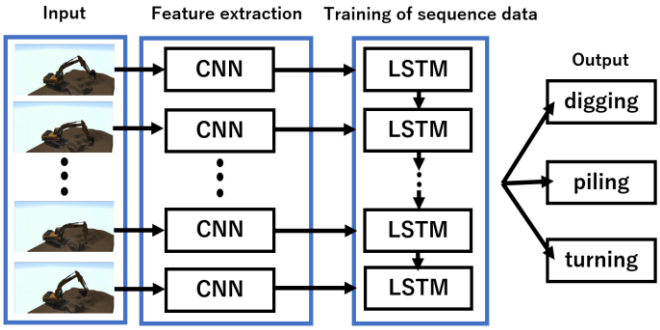

Figure 6. Learning model used in our proposed method consisting of a CNN coupled with LSTM.

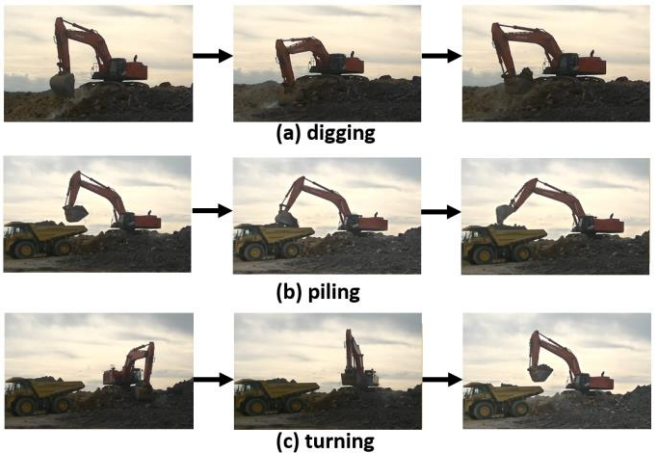

Figure 7. Examples of excavator action classes: (a) Digging; (b) Piling; (c) Turning.

D. Learning Model

The used network framework is illustrated in Fig. 6. It is a Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). The CNN is made out of a convolutional middle layer and pooling layer. It serves to extract a feature map with spatial information. LSTM has been developed for time-series data and is suited for learning long-term time dependencies. This combination allows recognition of construction machinery considering both spatial and temporal information.

The proposed network architecture is shown in Fig. 6. After extracting each RGB frame of a training data video, reduction to a size of 298×298×3 is conducted. This is the input to the CNN. The CNN used in this study was a pretrained Inception V3 [13]. The output from the CNN is then fed into the LSTM, consisting of 3 layers and classification is done by a softmax layer.

Figure 8. Example of a video segment collected at an actual construction site in Motomiya, Fukushima, Japan.

III. EXPERIMENTS

We conducted experiments on action recognition for three actions classes, digging, piling, and turning, using an excavator, which is one of the most used construction machinery at construction sites [14]. Each action class is shown in Fig. 7.

The training data required to train the excavator's action recognition was generated from four viewpoints using Vortex Studio simulator and about 60 videos segments were generated for each action class. Each video segment has a resolution of 1920×1080 and a frame rate of 30 fps. The average video duration is 7s, with the shortest being 4s and the longest being 13s. Using this training data, CNN and LSTM networks were trained for 150 epochs using batch size 32 with Adam optimizer.

As for the test data, three test datasets were created. The first test dataset was generated from the simulator to verify the accuracy of action recognition in the same domain. 20 video segments were generated for each action class. The second test dataset is a dataset collected using a remotely controlled scale model excavator, as initially reported in [11]. At that time, a background environment similar to the simulation environment was created. The second test dataset was also generated with 20 video segments for each action class. The third test dataset is a dataset collected at an actual construction site. We filmed a video of an excavator loading a dump truck at a construction site in Motomiya, Fukushima, Japan, and generated 30 video segments for each action class. All videos in those three test datasets had a resolution of 1920×1080 and a frame rate of 30 fps. Fig. 8 shows a sample of the data obtained at the construction site.

All filters used in this study were created using OpenCV [15] that is a highly optimized library with focus on real time applications.

The threshold parameter T was manually set to 100.

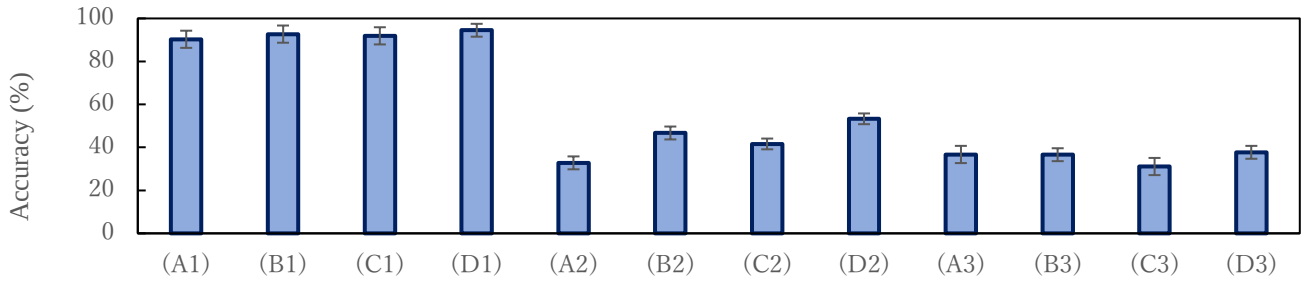

Figure 9. Average testing performance over 3 training runs of (A1) CNN+LSTM on simulator testing data, (B1) proposed Grayscale filter method on simulator testing data, (C1) proposed Edge filter method on simulator testing data, (D1) proposed filter $L^*(x, y)$ method on simulator testing data, (A2) CNN+LSTM on test data by remotely controlled scale model excavator and (B2) proposed Grayscale filter method on test data by remotely controlled scale model excavator, (C2) proposed Edge filter method on test data by remotely controlled scale model excavator, (D2) proposed filter $L^*(x, y)$ method on test data by remotely controlled scale model excavator, (A3) CNN+LSTM on test data by excavator and (B3) proposed Grayscale filter method on test data by excavator, (C3) proposed Edge filter method on test data by excavator, (D3) proposed filter $L^*(x, y)$ method on test data by excavator. 3 action classes were considered. Error bars correspond to one standard deviation.

The following experiments were conducted:

- (A1) CNN+LSTM trained on simulator generated training data and tested on simulator generated test data.
- (B1) The Grayscale filter method trained on simulator generated training data and tested on simulator-generated test data.
- (C1) The Edge filter method trained on simulator-generated training data and tested on simulator generated test data.
- (D1) The filter $L^*(x, y)$ method trained on simulator-generated training data and tested on simulator generated test data.
- (A2) CNN+LSTM trained on simulator generated training data and test data on remotely controlled scale model excavator.
- (B2) The Grayscale filter method trained on simulator generated training data and test data on remotely controlled scale model excavator.
- (C2) The Edge filter method trained on simulator-generated training data and test data on remotely controlled scale model excavator.
- (D2) The filter $L^*(x, y)$ method trained on simulator-generated training data and test data on remotely controlled scale model excavator.
- (A3) CNN+LSTM trained on simulator generated training data and test data on data from an actual construction site.
- (B3) The Grayscale filter method trained on simulator generated training data and test data on data from an actual construction site.
- (C3) The Edge filter method trained on simulator-generated training data and test data on data from an actual construction site.
- (D3) The filter $L^*(x, y)$ method trained on simulator-generated training data and test data on data from an actual construction site.

The calculation of accuracy was evaluated as the ratio of the number of correctly classified samples n_{correct} over the total number of samples in dataset N_{samples}.

$$\text{accuracy} = \frac{n_{\text{correct}}}{N_{\text{samples}}} * 100. \quad (3)$$

IV. Results and Discussions

Results regarding action recognition performance are reported on Fig. 9.

In (A1), (B1), (C1), and (D1), corresponding to cases where the training data and test data were both generated in the simulator, the accuracy of action recognition was 90.3%, 92.7%, 91.9%, and 94.5%, respectively. From this result, it can be seen that action recognition is successful if the training and test data domain are matched. Furthermore, the application of video filters yielded better performance. This is explained by the fact that the considered filters simplify the input video and act in that sense to reduce the dimensionality of the learning problem.

In (A2), (B2), (C2), and (D2), corresponding to cases where training was conducted with data from a simulator and testing was conducted with a remotely operated scale model excavator, the accuracy of action recognition was 32.8%, 46.7%, 41.6%, and 53.3%, respectively. It can be first noticed that all suffer drop in performance. The accuracy of (A2) before applying the video filter is similar to that of randomly classifying three classes, so it can be concluded that the action recognition has failed.

On the other hand, in (B2), (C2), and (D2), while still showing a performance drop, the introduction of video filters managed to significantly perform better than a random classifier. The filter $L^*(x, y)$ showed the highest accuracy among the three video filters, allowing a performance gain of over 20%. This shows that this filter was successful in bringing the training data and test data onto closer domains. In addition, we think that the filter $L^*(x, y)$ showed the highest accuracy for action recognition because it minimized unnecessary features and extracted only the necessary ones. The second highest accuracy was the Grayscale filter. The Grayscale filter was able to eliminate the color difference between the simulator training data and the scale model test data but it could not show higher accuracy because the domains were still different and we think it still contained a lot of unnecessary information for action recognition. Among the three filters, the Edge filter, which has the lowest accuracy while having removed most of the unnecessary features, is considered to have low accuracy due to having extracted erroneous edges in the blank background.

In (A3), (B3), (C3), and (D3), corresponding to cases where the training data was from a simulator and test data was from an actual construction site, the accuracy of action recognition was 36.7%, 36.6%, 31.1%, and 37.7%, respectively. For the test dataset created at the actual construction site, it can be seen that action recognition failed even after applying the video filters. Unlike the test dataset made using the remotely controlled scale model excavator, which had a background environment similar to the one in the simulation environment, this third test dataset contained various visual background noises, such as rocks falling and rising dust. It is thought that action recognition failed due to the influence of these noises. Reducing the susceptibility of our learning model to background variations could be considered by the addition of noise in the training data during training.

V. Conclusion

In our previous study, we proposed a method to perform action recognition of construction machinery using video filters from the training data generated by the simulator. In this study, in order to clarify the effect of the video filter, three video filters were used to determine which factors influence action recognition. We confirmed that the action recognition accuracy of the method with the filter $L^*(x, y)$ applied was 20% or more higher than the method without the video filter, and as a result, it was confirmed that for action recognition in different domains, is important to select filters that returns similar features while reducing the amount of unnecessary ones.

However, we obtained low accuracy for action recognition in our experiment with test dataset obtained at the actual construction site. This is thought to be caused by background variations, since the training data generated in the simulation excluded such noise from the background as much as possible. Therefore, in the future, we plan improve performance by generating training data with various noises in the background in the simulation environment in addition to pursuing the elaboration of a more suitable filter for action recognition.

References

[1] C.-F. Cheng, A. Rashidi, M. A. Davenport, and D. V. Anderson, "Activity analysis of construction equipment using audio signals and support vector machines," in Automation in Construction, vol. 81. Elsevier, 2017, pp. 240–253.

[2] S. Chi and C.H. Caldas, "Automated object identification using optical video cameras on construction sites," Journal of Construction Engineering and Management, 138(3), 2012, pp. 341–351.

[3] J. Kim, S. Chi, and J. Seo, "Interaction analysis for vision-based activity identification of earthmoving excavators and dump trucks," in Automation in Construction, vol. 87. Elsevier, 2018, pp. 297–308.

[4] M. Golparvar-Fard, A. Heydarian, and J. C. Niebles, "Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers," in Automation in Construction, vol. 27, no. 4, 2013, pp. 652–663.

[5] N. Pradhananga and J. Teizer, "Automatic spatio-temporal analysis of construction site equipment operations using gps data," in Automation in Construction, vol. 29. Elsevier, 2013, pp. 107–122.

[6] R. Akhavian and A. H. Behzadan, "Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers," in Advanced Engineering Informatics, vol. 29, no. 4. Elsevier, 2015, pp. 867–877.

[7] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in CVPR 2011. IEEE, 2011, pp. 3361–3368.

[8] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," in Image and vision computing, vol. 60. Elsevier, 2017, pp. 4–21.

[9] J. Kim and S. Chi, "Action recognition of earthmoving excavators based on sequential pattern analysis of visual features and operation cycles," in Automation in Construction, vol. 104. Elsevier, 2019, pp. 255–264.

[10] C. Chen, Z. Zhu, and A. Hammad, "Automated excavators activity recognition and productivity analysis from construction site surveillance videos," in Automation in Construction, vol. 110, 103045. Elsevier, 2020.

[11] J. Sim, J. Y. Louhi Kasahara, S. Chikushi, H. Yamakawa, Y. Tamura, K. Nagatani, T. Chiba, S. Yamamoto, K. Chayama, A. Yamashita, and H. Asama, "Action recognition of construction machineryfrom simulated training data using video filters," in Proceedings of the International Symposium on Automation and Robotics in Construction, 2020 (accepted).

[12] Vortex Studio, accessed 2020.08.11. https://www.cm-labs.com/vortex-studio/.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in CVPR 2016. IEEE, 2016, pp. 2818–2826.

[14] E. R. Azar and B. McCabe, "Part based model and spatial–temporal reasoning to recognize hydraulic excavators in construction images and videos," in Automation in Construction, vol. 24, 2012, pp. 194–202.

[15] OpenCV, accessed 2020.08.11. https://opencv.org/.