Action Recognition of Construction Machinery from Simulated Training Data Using Video Filters

Jinhyeok Sim[a], Jun Younes Louhi Kasahara[a], Shota Chikushi[a], Hiroshi Yamakawa[a], Yusuke Tamura[b], Keiji Nagatani[a], Takumi Chiba[c], Shingo Yamamoto[c], Kazuhiro Chayama[c], Atsushi Yamashita[a], and Hajime Asama[a]

[a]The University of Tokyo, Japan
[b]Tohoku University, Japan
[c]Fujita Corporation, Japan
E-mail: sim@robot.t.u-tokyo.ac.jp, louhi@robot.t.u-tokyo.ac.jp, chikushi@robot.t.u-tokyo.ac.jp
yamakawa@susdesign.t.u-tokyo.ac.jp, y.tamura@srd.mech.tohoku.ac.jp, keiji@i-con.t.u-tokyo.ac.jp
takumi.chiba@fujita.co.jp, syamamoto@fujita.co.jp, chayama@eae.co.jp
yamashita@robot.t.u-tokyo.ac.jp, asama@robot.t.u-tokyo.ac.jp

Abstract –

In the construction industry, continuous monitoring of actions performed by construction machinery is a critical task in order to achieve improved productivity and efficiency. However, measuring and recording each individual construction machinery's actions is both time consuming and expensive if conducted manually by humans. Therefore, automatic action recognition of construction machinery is highly desirable. Inspired by the success of Deep Learning approaches for human action recognition, there has been an increased number of studies dealing with action recognition of construction machinery using Deep Learning. However, those approaches require large amounts of training data, which is difficult to obtain since construction machinery are usually located in the field. Therefore, this paper proposes a method for action recognition of construction machinery using only training data generated from a simulator, which is much easier to obtain than actual training data. In order to bridge the feature domain gap between simulator-generated data and actual field data, a video filter was used. Experiments using a model of an excavator, one of the most commonly used construction machinery, showed the potential of our proposed method.

Keywords –

Action recognition; Deep learning; Video filter;

1 Introduction

In the construction industry, expenses related to heavy equipment, such as construction machinery, occupy large portions of the overall budget. Improving productivity and efficiency at construction sites is an important issue, and therefore, particular care has been attributed to the monitoring of such construction machinery. By obtaining and maintaining the time and costs required to complete a task, a more efficient construction plan can be made [1]. Traditionally, monitoring, consisting of recognizing and recording the actions performed by construction machinery, was conducted manually by the site manager's observations at the construction site [3]. This involved high costs and time. Therefore, automatic action recognition of construction machinery is highly desirable.

Figure 1. Simulator environment used in our proposed method (Vortex Studio [2])

Previous works dealing with the action recognition of construction machinery either employed added sensors onboard the machinery, such as GPS [4], or employed sensors positioned on the construction sites, such as cameras [5]. Approaches using cameras, which consist in placing several cameras in the construction site and recording the machinery at work, are especially appealing since they do not require modifications on the

construction machinery. Furthermore, inspired by the success of Deep Learning methods for Computer Vision-based approaches to human action recognition [6][7], several works have also managed remarkable results for the action recognition of construction machinery using Computer Vision and Deep Learning [8][9].

However, one of the major practical drawback of Deep Learning approaches is that a large amount of training data is required. Obtaining large amounts of training data is a tedious task. This was alleviated in some part for human action recognition thanks to the advent of the Internet and open-access data but it is not the case regarding specific targets such as construction machinery, which comes in a plethora of shapes and forms. High costs, in logistics, in manpower and in time, can be reasonably expected in order to obtain the training data appropriate for action recognition of construction machinery.

On the other hand, generating such training data using a simulator, illustrated in Figure 1, is comparatively easier: only a human operator to control the virtual construction machinery and a computer to run the simulator is needed. Therefore, the objective of the present paper is to conduct action recognition of construction machinery using Deep Learning based on training data obtained from a simulator.

Since the simulator differs too much from real construction sites to allow directly learning the features required for action recognition, a video filter is introduced in order to force a common ground between the data generated in the simulator and the data collected in actual construction sites.

2 Action Recognition of Construction Machinery based on Simulated Training Data

2.1 Concept

Training a model using training data generated from a simulator is not effective for action recognition of construction machinery at actual construction sites. This is because the training data would not be appropriate for the task at hand. This would be akin to train a model to distinguish pictures of dogs and cats and then testing it on pictures of cows. More generally, the features that could be extracted by the model from the feature space defined by the training data generated from the simulator do not match the features contained in the data collected at actual construction sites. Simply put, the construction machinery in the simulator does not *look like* real construction machinery.

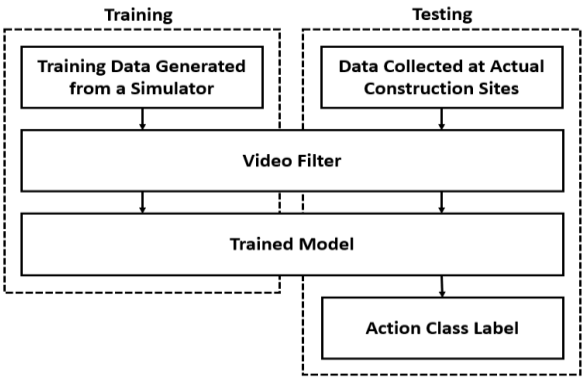

Figure 2. Overview of proposed method.

In such situation, a couple of options can be considered. Increasing the amount of training data, in a Big Data fashion, or a Data Augmentation approach could be considered. This would allow to expand the area of the feature space covered by the training data and hopefully encompass the desired portion of the feature space. However, in our case, this would have little hope to succeed since no matter the amount of additional training data generated from the simulator, the training data would still differ from the data collected at actual construction sites. Another option would be to attempt to bias the training data towards data collected at actual construction sites. To do so, domain adaption methods, i.e., using unlabeled data collected at actual construction sites in the training, or improvements to the simulator to match more closely actual construction sites can be considered. However, the former usually involves strong priors and expensive field data collection and the latter involves tedious software development.

The concept of the proposed method in this paper is a different approach: since the learning is hindered by a mismatch between the training data and the data collected at actual construction sites, the idea is to transform both of them into a third feature domain, which would be neither the feature domain of the simulator-generated training data nor of the data collected at actual construction sites. The introduction of a third feature domain would allow to bypass the previously mentioned issues related to trying to skew one domain towards another since the destination domain would be set independently of the available data.

An overview of the proposed method is shown in Figure 2. First, training data for action recognition of construction machinery is generated using a simulator. Then, both the training data generated from a simulator and the are transformed using a video filter. After this, the model for action recognition is trained on the transformed simulator training data and finally tested on the transformed data collected at construction sites.

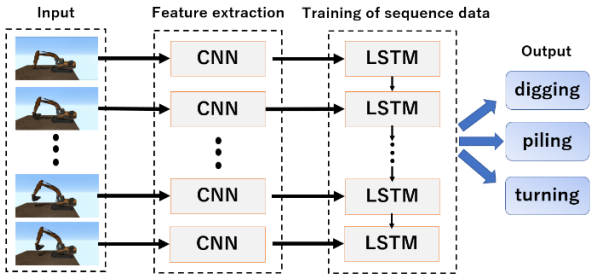

Figure 3. Learning model used in our proposed method consisting of a CNN coupled with LSTM.

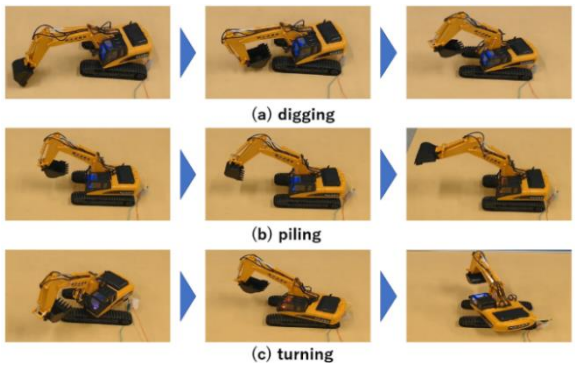

Figure 4. Examples of excavator action classes: (a) Digging; (b) Piling; (c) Turning

2.2 Training Data Generating using a Simulator

The most ideal training data for construction machinery action recognition is the data collected at actual construction sites, i.e., labeled data of the construction machinery working at the actual construction site. However, it is practically difficult to collect and label a large amount of data at the actual construction site. Therefore, in this study, Vortex Studio [2], which is a real-time simulator for mechanical system operation, is used to generate training data. A model of an excavator, one of the most commonly used construction machinery, is considered as shown in Figure 1.

In this study, RGB video data is used as input data, i.e., video of the construction machinery working. However, it is known that RGB data is easily affected by the background and camera viewpoint. Therefore, a background was created with only soil around the excavator. Moreover, the training data was generated from multiple camera viewpoints.

Concretely, to generate the training data, the camera viewpoint was first fixed while the excavator was moving in Vortex Studio and the excavator operation was conducted by human using a controller. The process was repeated several times with different camera viewpoints and different actions to generate training data.

2.3 Transformation to third Feature Domain using a Filter

In this study, the training data, generated from a simulator, differs from the data collected at actual construction sites. There is therefore a domain gap between the data used to train the model and the data used to test the model: the model trained in one domain cannot accurately conduct inference on the other domain. The concept of the proposed method is to match those two data on an independent third domain, where both would be similar.

The most obvious disparity between the data generated from a simulator and the data collected at construction sites is their appearance: they simply do not look alike and are easily differentiable. The differences mainly lie in color and texture: the simulator environment noticeably lacks the color modulations and textured appearance of surfaces compared to the real world. In order to erase those differences and basically make both the data generated from a simulator and the data collected at construction sites similar, an edge video filter is used. This edge filter is applied to both the training data generated from a simulator and the data collected at construction sites.

2.4 Action Recognition Learning Model

For action recognition, Deep Learning approaches such as using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) [10] or a method using 3D CNN [11] has been recently used, and these methods using deep learning have higher accuracy of action recognition than other methods.

In this study, CNN and LSTM are used as network framework. CNN is a network which middle layer is composed of convolutional layer and pooling layer and extracts a feature map containing spatial information. LSTM is a network suitable for time sequence data and capable of learning long-term dependency. Therefore, by using CNN and LSTM, it becomes possible to recognize the motion of the construction machinery that considers spatial information and temporal information at the same time.

The proposed network architecture is shown in Figure 3. First, from each training data video sample, recorded at 30 fps, each frame of RGB data is extracted. Next, the extracted RGB data is resized to a size of 298×298×3. Resized RGB data is inputted to the CNN and features are extracted. The CNN network used in this study uses a trained model called Inception V3 [12] pre-trained on over 1 million images. After that, the result of feature extraction from Inception V3 is inputted into LSTM. LSTM consists of three layers and classifies action labels in the softmax layer.

3 Experiments

In experiments, an excavator was selected as target for action recognition since it is one of the most commonly used construction machinery. Furthermore, action recognition was narrowed down to 3 action classes: digging, piling, and turning.

Training data was generated from four viewpoints using Vortex Studio simulator according to the procedure in Section 2.2. As a result, about 60 videos segments at 1920×1080 resolution and 30fps were generated for each action class. The average video duration is 7s, with the shortest being 4s and the longest being 13s.

The CNN and LSTM network in this study for action recognition was trained for 150 epochs using a batch size of 32 with the Adam optimizer.

Two test datasets were considered. The first test dataset was generated from the simulator with the same procedure as for the training data for the purpose of providing a baseline in ideal learning conditions. We generated about 20 video segments for each action class. The second test dataset corresponds to actual data collected at construction sites, i.e., real world test data. However, since we were unfortunately not able to gain access to actual construction sites, we opted instead to use a remotely controlled scale model excavator. At that time, we created a background environment similar to the simulation environment, then filmed the excavator work from four different angles and generated about 20 video segments for each action class.

Regarding the edge video filter, the sketch filter of the open source video editing software Shotcut [13] was used.

The following experiments were conducted:

• (A1) CNN+LSTM trained on simulator-generated training data and tested on simulator-generated test data.

• (B1) The proposed method trained on simulator-generated training data and tested on simulator-generated test data.

• (A2) CNN+LSTM trained on simulator-generated training data and tested on real world test data.

• (B2) The proposed method trained on simulator-generated training data and tested on real world test data.

The performance was evaluated by calculating the classification accuracy defined as the ratio of the number of correctly classified samples n_{correct} over the total number of samples in the test dataset N_{samples}.

$$\text{accuracy} = \frac{n_{\text{correct}}}{N_{\text{samples}}} * 100 \quad (1)$$

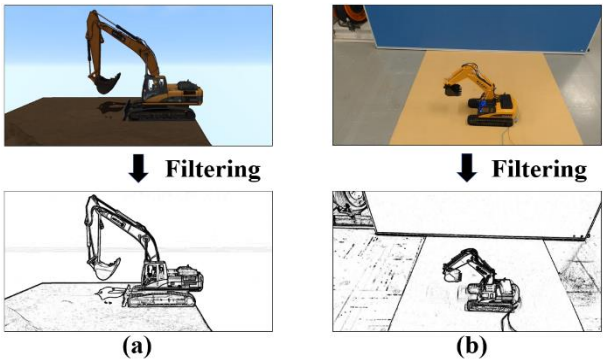

Figure 5. Effects of applying a video filter: both the data generated from a simulator (a) and real world data (b) have their differences suppressed.

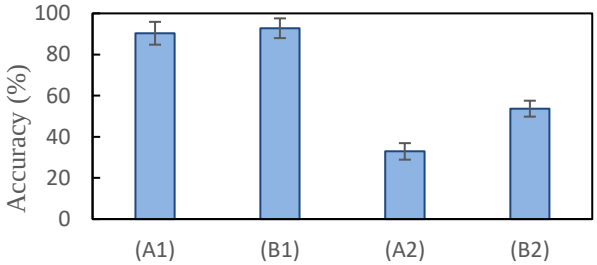

Figure 6. Average testing performance over 3 training runs of (A1) CNN+LSTM on simulator testing data, (B1) proposed method on simulator testing data, (A2) CNN+LSTM on real world test data and (B2) proposed method on real world test data. 3 action classes were considered. Error bars correspond to one standard deviation.

4 Results and Discussions

Figure 5 shows the simulator-generated data and the real world data along with the output after applying the filter. It can be noticed that most apparent features enabling differentiation have been successfully suppressed and that both data look very similar.

Results regarding action recognition performance are reported on Figure 6.

In experiments (A1) and (B1), when the training data and the test data are both generated from the simulator, it can be seen that both the proposed method and CNN+LSTM have high performance, exceeding 90%. This indicates that learning was successful. The proposed method achieved a slightly better performance at 92.3% accuracy. This is due to the fact that in our proposed method, the model learns on the training data on which the video filter was applied: this is simpler than learning directly from RGB.

In experiments (A2) and (B2), both CNN+LSTM and the proposed method were trained on the simulator-generated training data but testing was conducted with real world test data. It can be first noticed that both suffer

drop in performance. CNN+LSTM's accuracy dropped from 90.3% to 32.8%. Since 3 action classes were considered in our experiments, this is equivalent to random classification and it can be concluded that learning an appropriate model has failed. This illustrates the previously mentioned need for training data matching test data for successful learning of classification features in Section 2.1. On the other hand, the proposed method obtained an average performance of 53.7%. While there is still a performance drop, the proposed method managed to significantly perform better than a random classifier. This indicates that the introduction of a video filter allowed a performance gain of over 20%. This likely points out that the filter was successful in matching the simulator training data and real world data onto a similar domain.

5 Conclusion

A method to conduct action recognition of construction machinery from simulator-generated training data using a video filter was proposed. The differences between simulator data and real world data which prevented learning a successful model were suppressed by the use of a filter and allowed an accuracy increase of over 20%, effectively allowing the model to learn features for classification and not fail into a random classifier.

Experiments reported in this paper are still preliminary and served to demonstrate the potential of shifting the learning problem into a third domain. In the future, we plan investigate the effects of training data size and search for more suited filters for action recognition in order to improve performance. Incorporating the filter into the learning process, to learn a filter optimized for action recognition in parallel to the action recognition itself, is also considered.

References

[1] Jinwoo Kim, Seokho Chi, and Jongwon Seo: "Interaction Analysis for Vision-Based Activity Identification of Earthmoving Excavators and Dump Trucks," Automation in Construction, Vol. 87, pp. 297-308, 2018.

[2] Vortex Studio, accessed 2020.06.27. https://www.cm-labs.com/vortex-studio/

[3] Hyunsoo Kim, Changbum R.Ahn, David Engelhaupt, and Sanghyun Lee: "Application of Dynamic Time Warping to the Recognition of Mixed Equipment Activities in Cycle Time Measurement," Automation in Construction, Vol. 87, pp. 225-234, 2018.

[4] Nipesh Pradhananga, and JochenTeizer: "Automatic Spatio-Temporal Analysis of Construction Site Equipment Operations Using GPS Data," Automation in Construction, Vol. 29, pp. 107-122, 2013.

[5] Reza Akhavian, and Amir H.Behzadan: "Construction Equipment Activity Recognition for Simulation Input Modeling Using Mobile Sensors and Machine Learning Classifiers," Advanced Engineering Informatics, Vol.29, pp. 867–877, 2015.

[6] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng: "Learning Hierarchical Invariant Spatio-Temporal Features for Action Recognition with Independent Subspace Analysis," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), pp. 3361-3368, 2011.

[7] Samitha Herath, Mehrtash Harandi, and Fatih Porikli: "Going Deeper into Action Recognition: A Survey," In Image and vision computing, Vol 60, pp. 4-21, 2017.

[8] Jinwoo Kim, Seokho Chi: "Action Recognition of Earthmoving Excavators based on Sequential Pattern Analysis of Visual Features and Operation Cycles," Automation in Construction, Vol. 104, pp. 255-264, 2019.

[9] Chen Chen, Zhenhua Zhu, and Amin Hammad: "Automated Excavators Activity Recognition and Productivity Analysis from Construction Site Surveillance Videos," Automation in Construction, Vol. 110, 103045, 2020.

[10] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell: "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 2625-2634, 2015.

[11] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Tor-resani, and Manohar Paluri: "Learning Spatiotemporal Features with 3D Convolutional Networks," In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015), pp. 4489-4497, 2015.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna: "Rethinking the Inception Architecture for Computer Vision," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 2818–2826, 2016.

[13] Shotcut, accessed 2020.06.27. https://shotcut.org/