Automating the Estimation of Productivity Metrics for Construction Workers Using Deep Learning and Kinematics

Jacobsen E. L.[1], Teizer J.[2]
[1]Department of Civil and Architectural Engineering, Aarhus University, Denmark
[2]Department of Civil and Mechanical Engineering, Techincal University Denmark, Denmark
elj@cae.au.dk

Abstract. In this study, a novel method for direct work estimation is used to classify whether a painter is performing direct work or not. The aim is to build an accurate and reliable work classification algorithm that can help monitor construction sites. The method utilizes a deep learning algorithm using convolutional and long short-term memory layers to classify multivariate time-series data collected from five inertial measurement units (IMUs) mounted on the workers' arms, torso, and legs. Three models are developed, differing in window sizes from 3 seconds to 7 seconds. The best performing model achieves an accuracy of 90% and an F1-score of 87.6%. This is the first step towards a general model that can classify productivity measures for workers on construction sites, which will be a valuable input for monitoring construction sites and future analyses.

1. Introduction

The construction industry is very labor-intensive as it consists of many manual tasks. The ability to monitor and manage both workers and activities can be an essential tool to ensure projects are within the scope of time and cost. Several publications have over the last decade shown the construction industry lacks behind other industries in terms of productivity (Neve et al., 2020; Chapman et al., 2010). The conclusions vary due to the complex nature of the construction industry and the difficulty of consistently measuring productivity. The methods used are easily biased and often lack continuity, only looking at small percentages of the construction projects' timespan. No standardization has been made for monitoring productivity, which is currently done on several levels, from project to task and worker levels. Therefore, developing methods that allow for continuous monitoring of construction worker productivity is necessary to improve construction projects' performance.

Productivity can be an intangible concept as different tasks require different actions; different tasks have different definitions of whether an action is productive. In general, construction labor productivity (CLP) can be defined as the product output per person-hour worked. Several other factors are relevant when examining productivity, but as a measure of performance, CLP has been used alone as it is a significant factor for the total project cost, as well as because it is the only factor that is conscious of its contribution (Wandahl et al., 2021). This paper (1) gives an overview of current activity recognition and productivity estimation research in construction, (2) introduces a deep learning method for productivity estimation, (3) shows the early implementation and preliminary results of the model, and (4) discusses the future work regarding the method, the current challenges, and the potential.

2. Background

CLP has been extensively studied using manual methods, especially work sampling (Araujo et al., 2020; Wandahl et al., 2021; Gouett et al., 2011). Work sampling can be used to estimate CLP due to the direct correlation between the direct work (DW) category "producing" and

https://doi.org/10.7146/aul.455.c232

productivity. Automated processes, especially machine learning methods, have also been investigated to estimate productivity using static information such as project size, worker experience, or contractual agreements (Jacobsen and Teizer, 2022). The methods differ in the modality used for activity recognition, and most publications fall into one of three categories: images/videos, audio, or kinematic data (Sherafat et al., 2020).

2.1 Computer Vision Methods

Research within computer vision has seen an increase in interest from construction research (Jacobsen and Teizer, 2022). The methods based on images and videos have also been used for activity recognition. Luo et al. (2018) developed a convolutional neural network (CNN) model based on optical flow to recognize three actions: Walking, transporting, and steel bending. Support Vector Machine (SVM) is a highly used model, which has been used for activity detection both in lab settings (Khosrowpour et al., 2014) and on actual construction sites (Liu and Golparvar-Fard, 2015; Yang et al., 2016; Khosrowpour et al., 2014). Computer-vision methods have, in most cases, seen very well-performing models due to the efforts that have been put into the field of computer-vision methods, developing robust recognition models. However, computer-vision methods have limitations. Using cameras requires a well light scene, is a stationary method, and suffers in almost every configuration, especially in dynamic work settings, from occlusions (Jacobsen and Teizer, 2021). Furthermore, storing the information needed for the supervised machine learning algorithms can be costly as the files from cameras (pictures or videos) require large amounts of storage (Sherafat et al., 2020).

2.2 Audio-based Methods

Audio-based recognition for construction workers has primarily been used for activity detection through either an array of microphones or single microphones installed on job sites. A substantial part of the publications uses SVM for activity recognition. Several different features are used across the publications. Rashid and Louis (2020) utilize four domain-specific feature sets for their classification: time-, frequency-, cepstral-, and wavelet-domain features. Their algorithm was used to classify activities: nailing with nail-gun, hammering, table-saw cutting, and drilling. Others use only cepstral features (Yang et al., 2015) or only time features (Cheng et al., 2017). As the experiments differ in activities classified, experimental setup, and data quantities, it is impossible to compare these to each other, which is also why it is impossible to state which features are the most important when using audio-based methods for classifying activities. However, Rashid and Louis (2020), who use in total 318 features, shows that for their experiments, cepstral-domain features contribute the most to the algorithm's performance and wavelet-domain features contribute the least.

2.3 Kinematic-based Methods

Kinematic signals such as orientation data, magnetic fields, acceleration, and angular velocity have been used as a basis for activity detection algorithms. These data streams are collected from sensors, for instance, inertial measurement units (IMUs). This is possible as activities often have unique kinematic patterns. Different types of machine learning algorithms have been used to classify these patterns. Joshua and Varghese (2011) used decision trees, Naïve Bayes and multilayer perceptron to classify bricklaying activities using one acceleration sensor. More recent work has investigated other methods for activity classification, such as neural networks, KNN, and SVM (Akhavian and Behzadan, 2018; Ryu et al., 2018; Yang et al., 2019). Similarly to audio-based methods, SVM is a frequently used algorithm in this domain. This could be due

to its ease of implementation or the simplicity of most implementations. Some studies also utilize IMUs in combination with other datastreams, such as location (Cheng et al., 2013). Here the location is used for spatiotemporal reasoning, where zones have been predefined as working zones or material zones, to assist in classifying a movement into working, idling, walking or material handling.

3. Methods

In this section, an introduction to the proposed method for estimating direct work is given. First, the data acquisition and the pre-processing steps are explained. After introducing the initial steps, the deep learning model and the tuning investigated will be explained. Finally, the model performance will be assessed through accuracy, precision, and F1-scores.

3.1 Raw Kinematic Data and Pre-processing

The raw kinematic data is collected from five IMUs placed on the worker, as shown in Figure 1. The data is collected at 60 Hz, and six features are used for each IMU. The features are acceleration on three axes and angular velocity around three axes. 60 Hz is used for the data collection, as this is the standard output of the sensor. If the system was to be used for real-time prediction, the optimal frequency should be investigated, as this is believed to become a tradeoff between accuracy and computational efficiency. Other publications have used similar sampling frequencies (Sherafat et al., 2020)

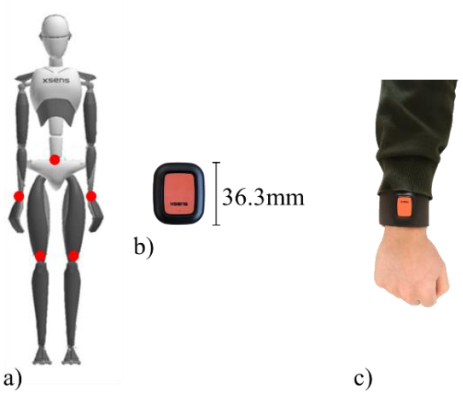

Figure 1: (a) Sensor position marked in red, (b) The sensor used for data collection, (c) Implementation on worker body

After data collection, each participants' data are combined into an array containing all features (30 in total). A fixed length sliding time window is used with 50% overlap as the input. Several window lengths have been investigated to assess how this affects performance. These configurations are window lengths of 3, 5, and 7 seconds. Each window is labelled based on video footage used as ground truth. The windows are labelled based on the amount of data that belongs to each class in the window, with the most frequent class being the overall label of the given window.

Data augmentation was implemented to increase the amount of data available for training the model. This method has been proven successful in many domains, such as activity recognition of construction machinery (Rashid and Louis, 2019). Two augmentations have been implemented: Jittering and scaling. Both of which have been done four times, with different distributions. An example of a 3 second window is shown in Figure 2.

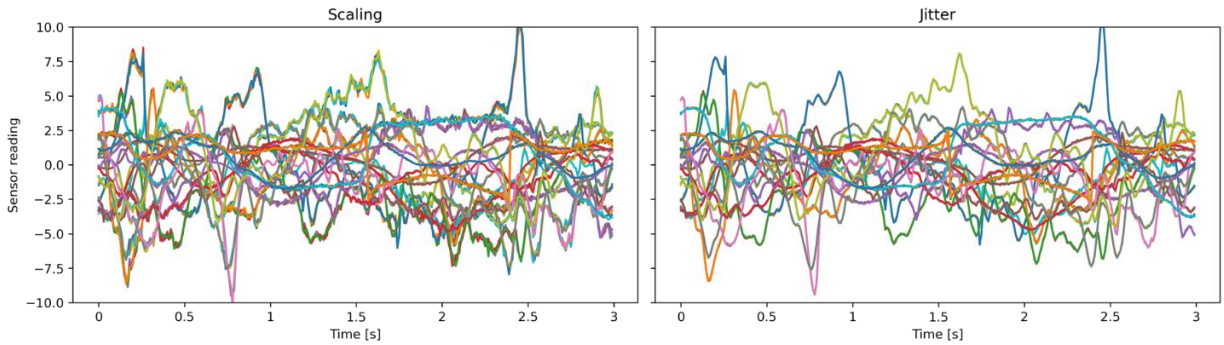

Figure 2: Data augmentation of a 3 second window. Each color represents a channel from the dataset (either acceleration or angular velocity).

3.2 Time Series Classification Model

The core of this method is a deep learning algorithm utilizing stacked convolutional layers to extract features and LSTM layers to find temporal relations in the feature maps created by the stacked convolutional layers. The convolutional layers use a 1D kernel to compute a temporal convolution of the data given by each sensor input. This kernel can be seen as a filter running over the input, filtering the data to find features. In stacked convolutional networks, the deeper layers will represent the features in more abstract ways, which have significantly impacted language processing (Krizhevsky et al., 2012) and computer vision (Jacobsen and Teizer, 2022). The method of combining convolutional layers with LSTM also has shown robust performance on human activity recognition before (Ordóñez and Roggen, 2016). The model used in this research consists of convolutional layers, each with 32 filters, whereafter the data is then reshaped into a 2D array fed into LSTM layers connected to dropout layers. Finally, the architecture uses a flatten layer followed by a dense layer before the output layer.

4. Model Evaluation and Case Study

The model evaluation is done on data of simulating painting of flat walls indoors in a research environment and could in the future act as pre-training for datasets collected in more realistic work environments. This gathered data is split into training and test data, where the training data was augmented, ultimately giving the model 8 times more data for training. Therefore, this model will allow for testing the robustness of the system before deploying it on a real work task settings on a construction site.

4.1 Data Collection

The dataset was collected in a laboratory environment, in which four sessions were completed and then combined into the final dataset. Data was in total collected for 90 minutes with the five sensors and a frequency of 60 Hz. This corresponds to 324.360 lines of data, each having 30 features. No additional features were calculated, meaning that only the features given directly from the IMU were used. The labelling was done using three classes, presented below:

1. Direct Work

   Direct work is work that physically adds value to the finished product. This creates the output previously discussed as one of the components of productivity. Examples of

direct work would be painting walls, laying tiles or bricks, assembling interior elements, and installing drywall sheets.

2. Indirect Work

   Indirect work is work that is necessary but does not directly contribute to the finished product. This could be activities such as getting materials and cleaning up the workstation from the predecessor's mess.

3. Waste

   Waste is work that is not necessary because it does not contribute to the finished product. This could include waiting, toilet visits, errands away from the construction site, walking to and from break, or walking between workstations without any tools.

The challenge with this classification is that these categories are not as tangible as the classification done in earlier activity recognition work, such as classifying whether a worker is hammering, walking, or lifting something.

The task was to paint walls around a predefined location, where a camera was set up to record the task. Breaks were done when it was seen as fitting. The breaks were labelled as waste and were spent on various tasks, such as walking around, getting a cup of coffee, or sitting and reading through the instructions. The direct work class was used whenever actual painting was happening. The indirect work class was used whenever the painting roller was dipped in the tray. An image from each of the three classes can be seen in Figure 3.

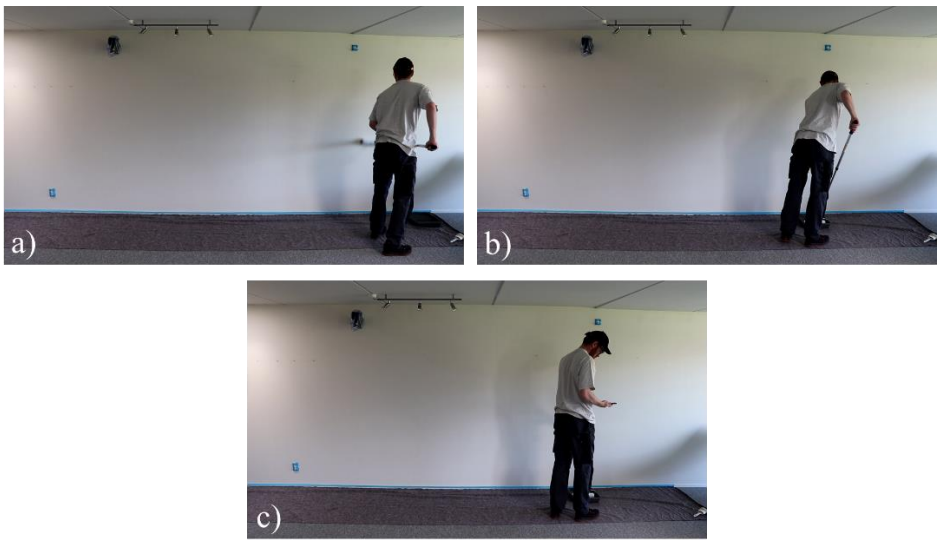

Figure 3: Ground truth recording of (a) Direct Work (painting the wall), (b) Indirect Work (picking up paint), and (c) Waste (answering incoming call).

The class distribution is shown in Table 1, where the number of occurrences for each class is shown.

Table 1: Distribution of training and test data given in windows of 3 seconds.

Class	Windows	Original training	Augmented training	Test data
Waste	1.734	1.273	10.184	461
Indirect work	587	449	3.592	138
Direct work	1.275	889	7.112	386

4.2 Implementation of Model

In the developed models, three different window lengths were used, the 7s model is shown in Figure 4, where also the labelling of the data is depicted.

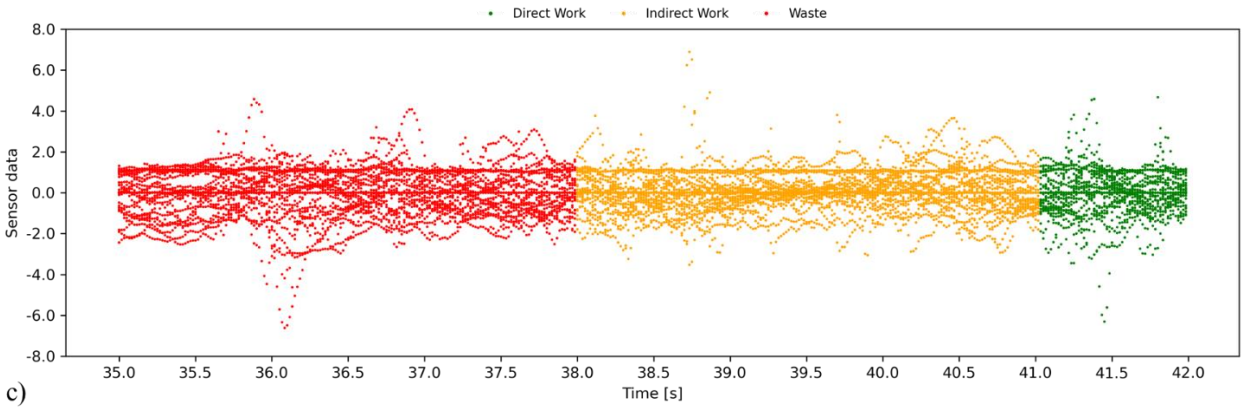

Figure 4: An example showing one of the 7s windows colored based on the label's class: green is direct work, yellow is indirect work, and red is waste

The project was developed on a windows machine using Tensorflow 2.6.0 in a Python 3.9.7 environment. For the three different models, each window consisted of an array of sizes 120 x 30, 300 x 30, or 420 x 30, respectively. The models were all set to train for 50 epochs and used a learning rate of 0.001. Minor hyperparameter tuning was done to the network, specifically the number of filters on the convolutional layers, the number of convolutional layers, the dropout, and the learning rate.

4.2 Evaluation Measures

Four performance measures are used to evaluate the performance of the developed models: accuracy, precision, recall, and F1 score. Accuracy only looks if the instances are classified correctly, whereas precision and recall look into how the misclassifications are distributed. In multiclass classification, a generalization of the measures is needed (Sokolova and Lapalme, 2009) to calculate the precision (Equation 1) and recall (Equation 2). Both measures are essential for a high performing model. However, it is often challenging to maximize both measures at once. Because of this, the F1-score is often used as a combination of the two (Equation 3). In the equations tp_i is true positive for the class C_i, fp_i is false positive, fn_i is false negative, and tn_i is true negative.

$$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fp_i)} \tag{1}$$

$$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fn_i)} \tag{2}$$

$$\frac{2 \cdot Precision(C_i) \cdot Recall(C_i)}{Precision(C_i) + Recall(C_i)} \tag{3}$$

5. Results

For the best performing model (7s), a model training history is shown in Figure 5. The model training history is depicted for the training and validation set, where the validation dataset is taken as 20% of the training dataset. As the dataset is very simple, only consisting of a worker painting a wall, it is expected that convergence will happen quickly. As seen in Figure 5, this is the case, as the model very quickly reaches high accuracy and minimal loss.

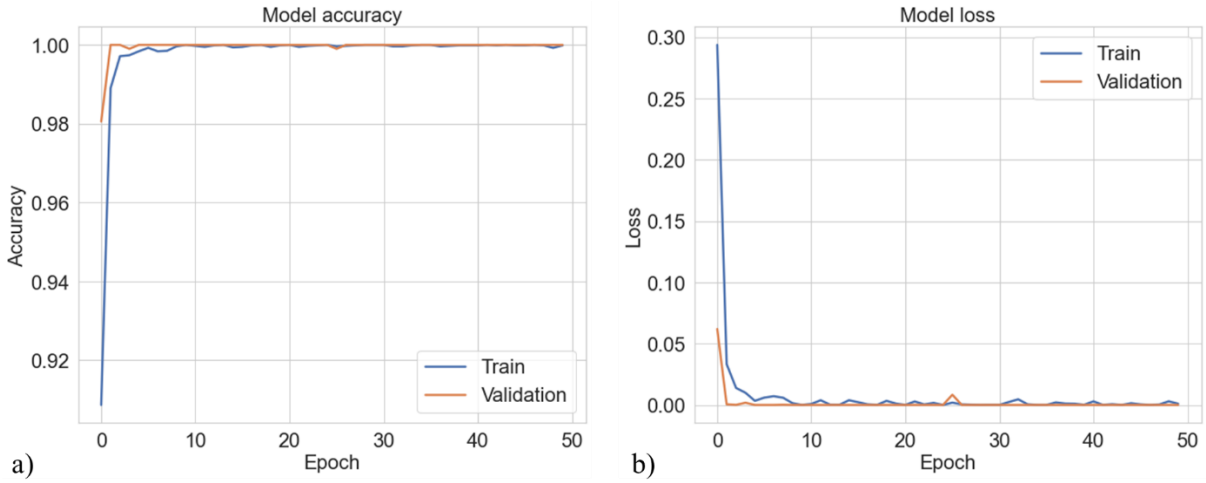

Figure 5: (a) Accuracy and (b) loss for 7s window model.

A confusion matrix is used for each of the three models to depict how well the model classifies and where it is mistaken (Figure 6).

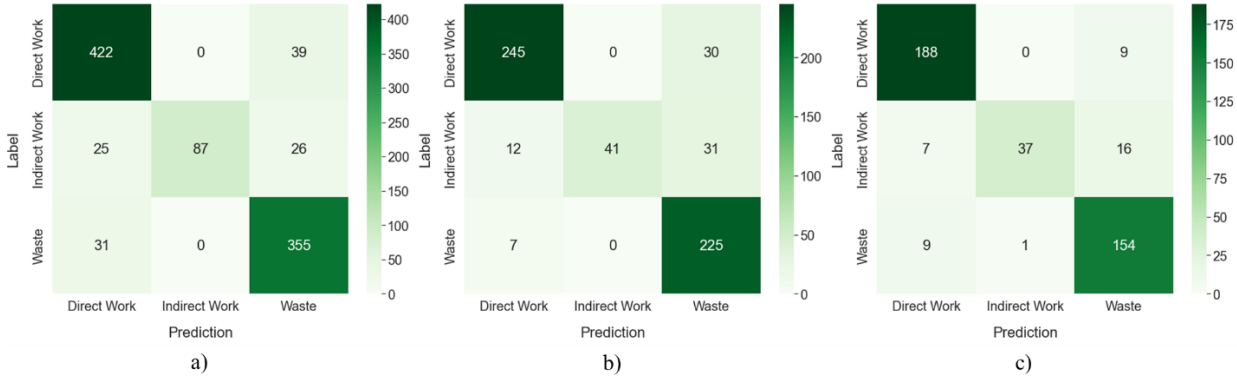

Figure 6: Confusion matrix for (a) 3s, (b) 5s, and (c) 7s windows.

All models perform satisfactorily, with the 7s model being the best among them. The accuracy ranges from 88.5% to 90%, as displayed in Table 2, where all performance measures can be seen. A general confusion from all models is with the class Indirect Work due to the imbalanced dataset. For the 3 second model, only 16.3% of the windows are in the Indirect Work class. This class imbalance could be solved by augmenting the worst represented class more than the others. All models have high precision in the Indirect Work category (100%, 100%, 97.4%, respectively), meaning that when the models predict Indirect Work, it is often or always true. The recall of Indirect Work however, is low (63%, 48.8%, 61.7% respectively). This means that whenever a worker is actually doing direct work, the model will misclassify the action in 37% to 51.2% of the time, depending on the model. All models perform well for Direct Work and Waste, with most windows being predicted to the true label and therefore having high

accuracy, recall and precision. A substantial part of the windows misclassified are windows where two or even three classes are present in the same window. An initial step to increase model performance could be to exclude windows where the class distribution is above a threshold.

Table 2: Evaluation measures for deep learning models.

Model	Accuracy	Precision	Recall	F1-Score
3s window	0.885	0.909	0.822	0.863
5s window	0.865	0.905	0.767	0.885
7s window	0.900	0.919	0.837	0.876

The three models evaluated are all successful solutions to classify the productivity measures, all being close to 90% accurate. This is seen as sufficient for estimating worker productivity, as other manual methods such as work sampling use far less frequent data collection for their assessment. If the current models' classification was to be downsampled to giving one classification each minute, it is expected that this model would produce a more accurate picture of the construction site than the manual methods can.

6. Future Steps

As more trades become part of the project, it is expected that the model's performance will decrease. To combat this, hyperparameter tuning on all the individual models will be done, and new deep learning methods will be investigated. Methods such as Graph Convolutional Networks and attention-based models have seen successful implementations for timeseries problems (Zerveas et al., 2021; Heidari and Iosifidis, 2021) and could lead to robust performances for models classifying multiple trades.

The model's output is seen as a valuable input for decision-makers, and therefore, it is necessary to investigate the possibility of near real-time outputs. Having real-time outputs would make the model more valuable for the industry, as decisions could be made based on the current output of the model. Furthermore, the number of classes could be extended to distinguish between the activities, as the literature has several subgenres to the classes used in this paper. To enable real-time outputs, feature importance should be investigated, to examine what features are important and which could be left out. This would not only lead more efficient computations, but could make the setup less invasive on the worker.

In the results section, the misclassification was discussed to primarily happen in windows where two or three classes are present. Implementing a dynamic window would make this situation impossible and make the model more robust.

7. Conclusion

Estimating productivity automatically provides a more robust solution than current methods such as work sampling. CLP is a important metric for understanding a construction project. To be able to estimate the metric automatically using simple kinematics sensors allows for greater insights and allows for detailed analyses, which potentially could lead to a higher CLP. This paper provides a trained classification model based on deep learning methods, utilizing kinematics data from IMUs. The paper evaluates the effect of window sizes on the models to

evaluate this hyperparameter to increase performance. Currently, the method has only been trained and tested on one task but shows good performance. Further analysis will determine how well the model generalizes to other tasks and trades.

The hyperparameter tuning of this research was minimal, and future research should focus on finding optimal parameters to enable the models to generalize. As the data is collected in a controlled environment, data collection on a construction site is planned for future evaluation of the method. Creating a high-quality kinematics dataset of human activities on construction sites is an important step for the field, as this is currently unavailable. This would also enable an evaluation of the robustness of the algorithms, enabling validation of their output in a real construction setting. The data used in this paper would be utilized for pre-training on models that should be deployed on a real construction site, which has been shown to increase the performance of models in the past (Zerveas et al., 2021).

To better understand how the model functions, it would be beneficial to use local interpretation methods to get local explanations for the classifications done by the model. This could potentially lead to pruning down the model and exclude sensors. The pruning would make for a more efficient model, as fewer computations would be needed. If the model is to be used in a real-time environment, this is seen as an essential step. The exclusion of sensors would also be beneficial, as this would mean less intrusion on the worker and make the setup less invasive.

References

Akhavian, R., Behzadan, A. H. (2012). Remote Monitoring of Dynamic Construction Processes Using Automated Equipment Tracking, *Construction Research Congress 2012*, 10.1061/9780784412329.137.

Araujo, L. O. C., Neto, N. R., Caldas, C. (2020). Analyzing the Correlation between Productivity Metrics, *Construction Research Congress 2020*, 10.1061/9780784482889.077.

Chapman, R. E., Butry, D. T., Huang, A. L. (2010). Measuring and improving U.S. construction productivity, *Proceedings of the 2010 CIB World Congress.*

Cheng, C. F., Rashidi, A., Davenport, M. A., Anderson, D. V. (2017). Acoustical Modeling of Construction Jobsites: Hardware and Software Requirements, *Computing in Civil Engineering*, 10.1061/9780784480847.044.

Cheng, T., Teizer, J., Migliaccio, G. C., Gatti, U. C. (2013). Automated task-level activity analysis through fusion of real time location sensors and worker's thoracic posture data, *Automation in Construction*, 10.1016/j.autcon.2012.08.003.

Gouett, M. C., Haas, C. T., Goodrum, P. M., Caldas, C. H. (2011). Activity Analysis for Direct-Work Rate Improvement in Construction, *Journal of Construction Engineering and Management*, 10.1061/(ASCE)CO.1943-7862.0000375.

Heidari, N., Iosifidis, A. (2021). Temporal Attention-Agumented Graph Convolutional Network for Efficeient Skeleton-Based Human Action Recognition, *Proceedings of ICPR 2020 - 25th International Conference on Pattern Recognition (ICPR)*, 10.1109/ICPR48806.2021.9412091

Jacobsen, E. L., Teizer, J. (2022). Deep learning in construction: Review of applications and potential avenues, *Journal of Computing in Civil Engineering*, 10.1061/(ASCE)CP.1943-5487.0001010.

Jacobsen, E. L., Solberg, A., Golovina, O., Teizer, J. (2021). Active personalized construction safety training using run-time data collection in physical and virtual reality work environments, *Construction Innovation*, 10.1108/CI-06-2021-0113.

Joshua, L., Varghese, K. (2011). Accelerometer-Based Activity Recognition in Construction, *Journal of Computing in Civil Engineering*, 10.1061/(ASCE)CP.1943-5487.0000097.

Khosrowpour, A., Niebles, J. C., Golparvar-Fard, M. (2014). Vision-based workface assessment using depth images for activity analysis of interior construction operations, *Automation in Construction*, 10.1016/j.autcon.2014.08.003.

Liu, K., Golparvar-Fard, M. (2015). Crowdsourcing construction activity analysis from jobsite video streams, *Journal of Construction Engineering and Management*, 10.1061/(ASCE)CO.19437862.0001010.

Michel, P., Levy, O., Neubig, G. (2019). Are sixteen heads really better than one?, *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, arXiv:1905.10650.

Neve, H. H., Wandahl, S., Lindhard, S., Teizer, J., Lerche, J. (2020). Determining the Relationship between Direct Work and Construction Labor Productivity in North America: Four Decades of Insights, *Journal of Construction Engineering and Management*, 10.1061/(ASCE)CO.1943-7862.0001887.

Ordóñez, F. J., Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition, *Sensors*, 10.3390/s16010115.

Rashid, K. M., Louis, J. (2019). Times-series data augmentation and deep learning for construction equipment activity recognition, *Advanced Engineering Informatics*, 10.1016/j.aei.2019.100944.

Rashid, K. M., Louis, J. (2020). Activity identification in modular construction using audio signals and machine learning, *Automation in Construction*, 10.1016/j.autcon.2020.103361.

Ryu, J., Seo, J., Lee, S. (2019). Automated Action Recognition Using an Accelerometer-Embedded Wristband-Type Activity Tracker, *Journal of Construction Engineering and Management*, 10.1061/(ASCE)CO.1943-7862.0001579.

Sainath, T. N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A. R., Dahl, G., Ramabhadran, B. (2015) Deep convolutional neural networks for large-scale speech tasks, *Neural Networks*, 10.1016/j.neunet.2014.08.005.

Sherafat, B., Ahn, C. R., Akhavian, R., Behzadan, A. H., Golparvar-Fard, M., et al. (2020). Automated Methods for Activity Recognition of Construction Workers and Equipment: State-of-the-Art Review, *Journal of Construction Engineering and Management*, 10.1061/(ASCE)CO.1943-7862.0001843.

Sokolova, M, Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks, *Informaiton Processing and Management*, 10.1016/j.ipm.2009.03.002.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al. (2017). Attention is all you need, *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*, ArXiv:1706.03762.

Wandahl, S., Pérez, C. T., Salling, S., Neve, H. H., Lerche, J., Petersen, S. (2021). The Impact of Construction Labour Productivity on the Renovation Wave, *Construction Economics and Building*, 10.5130/AJCEB.v21i3.7688.

Yang, J, Shi, Z., Wu, Z. (2016). Vision-based action recognition of construction workers using dense trajectories, *Advanced Engineering Informatics*, 10.1016/j.aei.2016.04.009.

Yang, S., Cao, J., Wang, J. (2015). Acoustics recognition of construction equipments based on LPCC features and SVM, *34th Chinese Control Conference*, 10.1109/ChiCC.2015.7260254.

Yang, Z., Yuan, Y., Zhang, M., Zhao, X. (2019). Assessment of Construction Workers' Labor Intensity Based on Wearable Smartphone System, *Journal of Construction Engineering and Management*, 10.1061/(ASCE)CO.1943-7862.0001666.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C. (2021). A Transformer-Based Framework for Multivariate Time Series Representation Learning, *27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, 10.1145/3447548.3467401.