2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

ActDNN: INDEPENDENT AND SEQUENTIAL LEARNING FRAMEWORK FOR ACCURATE CONSTRUCTION EQUIPMENT MONITORING

Sneha Verma[1], Wahib Saif[2], Xiang Xie[1], and Mohamad Kassem[1]
[1]Newcastle University, Newcastle Upon Tyne, United Kingdom
[2]Northumbria University, Newcastle Upon Tyne, United Kingdom

Abstract

This paper presents multi-label ActDNN, a novel neural network for activity recognition on construction sites, addressing limitations in vision-based methods reliant on large structured datasets. ActDNN facilitates robust multi-label activity recognition through independent learning and sequential learning. In independent learning, the network was trained and tested on an independent set of frames, achieving an accuracy of 99.82%. In sequential learning, sequential information was utilised to predict the sequential activities of an excavator and two trucks, achieving prediction accuracies of 97.79%, 89.67%, and 86.48%, respectively. This study enhances vision-based methods for automating sequential activity and productivity analysis, offering scalable and efficient construction equipment monitoring.

Introduction

Construction sites are highly dynamic environments where various types of equipment operate under interdependent workflows. Effective equipment management is crucial for maintaining productivity, optimising resource utilization, and minimising operational inefficiencies (Behzadan et al. 2008; Kamat et al. 2011; Yang et al. 2019). Beyond operational efficiency, tracking and monitoring construction equipment is critical in achieving sustainability goals by enabling emissions tracking, optimising fuel consumption, and supporting decarbonisation strategies (Saif et al. 2025). Additionally, equipment tracking enhances project progress monitoring, ensuring alignment with planned schedules, reducing delays, and improving overall project control. Despite the critical importance of equipment monitoring, existing methods remain inadequate. Traditional manual tracking is labor-intensive, error-prone, and inefficient, particularly on large-scale projects (Saif & Alshibani, 2024). While telematics-based solutions offer automated data collection, they often fall short in mixed equipment fleets due to interoperability issues, inconsistent data integration, and limited real-time analytics (Rogage et al., 2022). These limitations hinder proactive decision-making, reducing opportunities for performance optimisation and sustainability improvements.

To address these challenges, this study aims to develop an automated approach for recognising equipment activities and estimating productivity on construction sites. It proposes and tests an innovative neural network model that combines independent and sequential learning to accurately classify equipment activities. By automating these processes, the study enhances real-time monitoring to optimise equipment utilization and provides reliable productivity estimates, supporting more efficient and sustainable construction operations.

Related Studies

Tracking and controlling construction operations is essential for successful project delivery. It optimises resource use, minimises delays, and enhances productivity. Effective monitoring helps identify inefficiencies and detect issues early, enabling prompt corrective actions (Saif & Alshibani, 2024). Traditional productivity estimations through manual on-site inspection are widely recognized by site managers as inefficient and ineffective (Kim et al 2019; Kim and Chi 2020). This human-dependent approach is also resource-intensive, posing significant demand on labour, time and cost (Kim et al. 2016, 2018, 2017). These limitations hinder the opportunity to improve equipment's operational efficiency, which is considered one of the major productivity blind spots in infrastructure projects (Kim et al. 2019; Kim and Chi 2020, Kassem et al., 2021). Major organisations in the UK, such as National Highways, which oversee massive infrastructure projects, have acknowledged the innovative use of camera systems in construction for enhancing safety and monitoring progress. Such recognitions, often implemented by their contractors, highlight the industry's move towards integrating surveillance technology to improve site management and safety. However, solutions for monitoring equipment productivity remain limited in practice. The prevalent method is mainly based on telematics embedded by Original Equipment Manufacturers (OEM), which are not consistent across

mixed equipment fleets, and their data is not promptly available to decision-makers (e.g., general contractors, subcontractors, clients) (Kassem et al., 2021). To address these limitations, this study develops a video-based method that automatically provides the desired level of semantic and temporal details to monitor various earth-moving equipment (e.g., excavators and up to three dump trucks) and the conducted activity (e.g., loading, swinging, idling, moving, and dumping). The proposed in-house algorithm has been validated to effectively model equipment behaviour on construction sites with minimal human intervention.

Problem formulation

Once cameras are installed on sites, the challenge emerges to automatically analyse the activities of construction equipment from video data, which entails four sub-problems: a) Locating and classifying construction equipment within individual video frames; b) Associating detections across video frames to identify individual equipment (e.g., one excavator and three dump trucks) trajectories; c) Determining the specific activity being performed by each piece of equipment at each frame; and d) Predicting the ongoing activity based on the camera video frames. The first three sub-problems are collectively referred to as *independent learning*, as they do not account for dependencies between data points on temporal sequences. The final sub-problem, on the other hand, is addressed through *sequential learning*, which focuses on understanding and predicting activity patterns over time (e.g., the temporal order of frames to recognise patterns, transitions, or trends).

Several challenges affect the video dataset, primarily stemming from the variability of heavy equipment. Differences in models and sizes contribute to significant intraclass variability, complicating detection and classification. Additionally, the visual appearance of equipment can be altered by factors such as dirt, shadows, and partial occlusion. The dynamic nature of construction sites further adds complexity, with cluttered backgrounds and fluctuating lighting conditions making it difficult to distinguish equipment from its surroundings, and sometimes equipment is partially or fully obscured by other objects or structures, posing a substantial challenge to accurate detection (Roberts and Golparvar-Fard, 2019).

Proposed ActDNN method

The framework for ActDNN is developed for video activity recognition, utilising Convolutional Neural Networks (CNN) and ResNet-18 (He et al. 2016) to accurately classify multiple activities per frame. This approach leverages the spatial feature extraction capabilities of CNNs and the deep residual learning of ResNet-18 to effectively classify activities within the frames. Additionally, a transformer with a self-attention model has been integrated to capture the temporal dependencies and interactions between activities across frames, enabling a more comprehensive understanding of activity dynamics. The framework is trained on a public dataset from various site locations, ensuring robustness and generalisation of the algorithm as shown in Figure 1. The whole research has been performed on the 13th Generation Intel® Core™ i9-13950HX vPro® Processor Linux machine with 128 GB DDR5-5600 5600 MHz SODIMM memory and an NVIDIA® GeForce RTX™ 4090 with 16 GB GDDR6, with Python version 3.6 and Pytorch version 1.5.1.

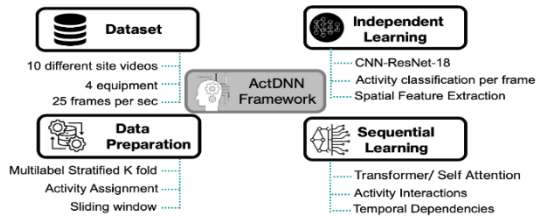

Figure 1: Shows the schematic of the framework of Act DNN.

Data Preparation

Data has been prepared to predict the activity of four pieces of heavy equipment (Excavator, $Truck_1$, $Truck_2$ and $Truck_3$) involved in the site activities. This involved organising and structuring the dataset, which included video frames with associated activities and bounding boxes for different pieces of equipment. Each frame in the dataset contains multiple activities conducted by different equipment, which are labelled based on specific frame intervals. The activities are represented by a start frame (f_s), and an end frame (f_e), and the activity types are A∈ {idle, swing bucket, load bucket, dump, and move}. These activities are associated with each frame (f), and the activities for a given frame can be expressed as $A(f) = \{A_i, A_j, \ldots\}$, where A_i and A_j represent the activities occurring in that frame for various pieces of equipment, which results in multiple labels per frame. For each frame, bounding boxes are extracted, where each bounding box corresponds to equipment (e.g., an Excavator or Truck) and is defined by its coordinates $(x_{min}, y_{min}, x_{max}, y_{max})$.

To start DNN, the activities for each frame are converted into a multi-label binary vector representation. Each element of the vector corresponds to an activity, with a value of 1 indicating that the activity is present in the frame and 0 indicating its absence. This can be mathematically represented as a vector, where activity is present and otherwise. This binary vector format enables efficient handling and processing of multi-label classification tasks. To split the data for machine learning, Multi-label Stratified Shuffle Split (MSSS) (Merrillees and Du 2021) is used, which ensures the distribution of activity labels across the splits (training, validation, and test sets) is similar to the overall dataset. This approach preserves the relative distribution of

activity labels in each subset, which is crucial for maintaining a balanced representation of activities during model training and evaluation. Hence, $X_{full} = \{x_1, x_2, \ldots, x_N\}$represent the set of all frames, where each sample x_i has a corresponding multi-label vector y_i that represents its activities, with the objective of maintaining the same distribution of activities in each subset. Mathematically, this can be expressed as:

$$P(y_{full}|X_{train}) \approx P(y_{full}|X_{full}) \quad (1)$$

where X_{train} the training set X_{full} is the entire frames, also representing the full multilabel vectors. The initial test size is defined as 20%. After this, the test set is further divided into 50% validation and 50% final test sets.

Finally, the data frames D_{train}, $D_{validation}$ and D_{test} were saved in CSV files that consist of frames associated with multi-label label annotations. Each frame x_i is paired with a label vector $y_i \in \{0,1\}C$ where $C = 5$ represents the total number of possible activities. This custom dataset class handles loading image files and their labels. Each image is located using a recursive search and transformed via a preprocessing pipeline. This pipeline includes resizing to 224×224, normalization using ImageNet statistics and conversion to a tensor $T(x_i)$. Labels are padded to a fixed length C, ensuring uniformity across samples for training the ActDNN.

Independent Learning

After resizing the image, the ResNet-18 was designed for multi-label classification. An additional convolutional layer is added in front of the ResNet backbone, transforming the input image $R^{32\times Height/2\times Width/2}$ as shown in Figure 2. The processed image is then passed through the ResNet-18. where the initial convolutional layer is updated to accept the 64-channel input. Finally, a fully connected (FC) layer outputs logits $\hat{y} \in RC$, where each element $\hat{y}_j$ corresponds to the unnormalized score for the j-th activity class. During the training process, Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) is used, as it is well-suited for multi-label classification. For a batch of size (total number of samples)N, it is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}[y_i[j]log\sigma(\hat{y}_i[j])] + (1 - yi[j])\log(1 - \sigma(\hat{y}i[j])) \quad (2)$$

where $\sigma(\hat{y}) = \frac{1}{1+e^{-\hat{y}}}$ is the sigmoid activation function is to ensure that each activity label is independently considered during loss computation.

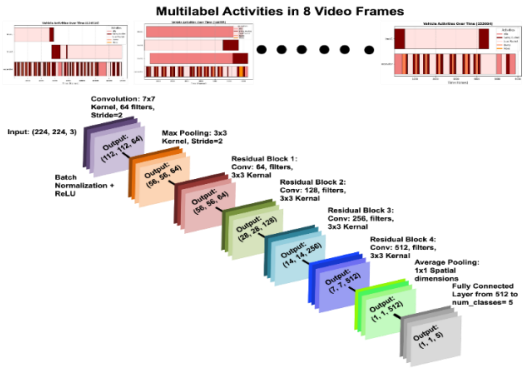

Figure 2: Architecture of ActDNN for independent Learning

To prevent overfitting during training, the loss function L is minimised using backpropagation. The model parameters are updated iteratively through the Adam optimiser, which is chosen for its efficiency and adaptability. A learning rate of 0.0001 is employed to ensure gradual error convergence. This approach facilitates stable training while promoting generalisation to unseen data. Hence, during each epoch, for a batch of images $\{x_i\}_i^B = 1$ and labels $\{y_i\}_i^B = 1$, the model computes the logits $\{\hat{y}_i\}_i^B = 1$ through forward propagation. Where ŷ is the predicted classes and $y_i$is the actual classes, B is the batch size and x_i is the input images. Gradients of the loss concerning model parameters are computed during the backward pass, and parameters are updated using:

$$\theta \leftarrow \theta - \eta \nabla\theta L \quad (3)$$

where η is the learning rate = 0.0001 and $\nabla_\theta L$ is the gradient of the loss with respect to θ, where θ is weights and biases of the neural network. Finally, training $(L_{train})$and validation loss (L_{val}) are computed at each epoch for monitoring convergence using the learning curve: $\left\{L_{train}^{(t)}, L_{val}^{(t)}\right\}T_t = 1$, where T is the total number of epochs as shown in Figure 3. In this figure, the purple bar shows the loss during the model training, which started from 0.068 and exponentially reduced up to 0.01 for epoch 13 and then got saturated. Whereas pink bars show the validation loss of the trained model, which started from 0.06 and follows the same trend and converges up to 17 iterations. As the model stopped learning further and converged, we used early stopping to stop further training to mitigate unnecessary computational overhead by halting the training once the model showed no significant gain in performance.
Finally, the accuracy (Acc) was calculated as 99.82% with the help of eq. 7 as follows:

$$Acc = \frac{1}{N\times C}\sum_{i=1}^{N}\sum_{j=1}^{C} \mathbb{1}_{\forall}([\sigma(\hat{y}i[j]) > 0.5] = yi[j]) \quad (4)$$

where $\mathbb{1}_{\forall}$ is the indicator function, and the sigmoid activation ensures classes are thresholded at 0.5, N and C is the total number of samples in the batch and total number of classes, respectively.

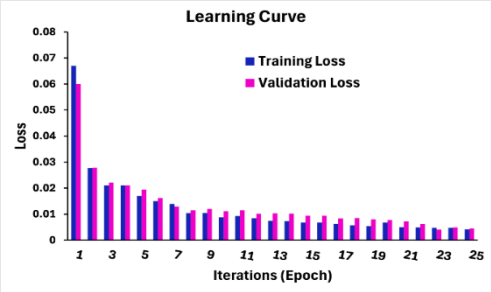

Figure 3: Learning curve for ActDNN

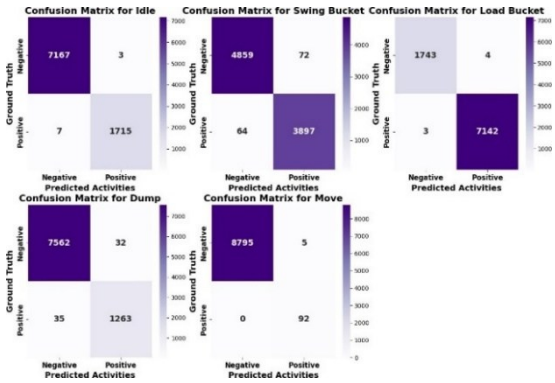

Figure 4: Confusion matrix for predicted actual activity

Figure 4 shows the confusion matrix, proving that the model performs well across all activity classes, with some variations in accuracy between them. For Idle and Move, the model shows high accuracy with a large number of true positives (TP) and very few false positives (FP) and false negatives (FN), suggesting it is highly effective in classifying these activities. Specifically, Idle has 7167 TP and only 3 FP, indicating that the model rarely misclassifies other activities as Idle. Similarly, move has 8795 TP and only 5 FP, showing excellent predictive performance. For other activities like swing bucket and load bucket, the model maintains good performance but shows slightly higher misclassification rates. Swing Bucket has 3897 TP, but there are 72 FP and 64 FN, indicating that the model sometimes confuses it with other activities, such as Idle. Likewise, the load bucket has 7142 TP but 4 FP and 3 FN, showing a small number of misclassifications. The dump activity shows the most room for improvement, with 1263 TP but 32 FP and 35 FN, indicating that the model struggles more to distinguish dump from other activities compared to others. This model demonstrates strong predictive power, with excellent precision and recall for most activities but it can further improve performance for less frequent activities like dumping by increasing the data size. Finally, Table 1 summarises the performance metrics (accuracy, precision, and recall) of the developed algorithm for different activities and equipment configurations such as Excavator E and Truck (T_1, T_2, T_3).

Table 1. Performance of the Model for Different Activities and Equipment Configurations. (E = Excavator, T1 = Truck1, T2 = Truck2, T3 = Truck3; Activities: 0 = Idle, 1 = Swing Bucket, 2 = Load Bucket, 3 = Dump, 4 = Move)

Activity	E	T_1, E	T_1, T_2, T_3, E	T_2, E
Accuracy				
0	99.82	99.90	100.00	99.94
1	99.53	97.95	98.88	98.70
2	99.82	99.92	100.00	100.00
3	99.82	98.97	99.36	99.48
4	99.94	100.00	100.00	100.00
Precision				
0	99.84	100.00	100.00	99.60
1	98.45	97.81	99.04	98.90
2	99.82	99.93	100.00	100.00
3	98.33	97.79	96.88	96.85
4	98.41	0.00	0.00	0.00
Recall				
0	99.69	97.16	100.00	100.00
1	99.48	98.23	98.72	97.97
2	99.64	99.98 0	100.00	100.00
3	96.72	96.62	98.94	99.54
4	100.00	0.00	0.00	0.00

From this table, the performance for various combinations of equipment can be visualized. For example, the model achieves high accuracy (99.82%) for the idle activity (0) across all configurations, with slight variations in precision and recall. Notably, precision and recall for activity 4 (move) are lower in some configurations, indicating challenges in accurately detecting this activity, particularly in the case of T_1, T_2, T_3, and E.

In short, independent learning can be highly effective when activity identification is based solely on individual frames, particularly in construction environments with random and irregular image sequences. This approach focuses on the specific characteristics and activities of the equipment present at the site, allowing for the prediction of activity based on isolated frames without considering the context of the background. One of the key advantages of this method lies in its robustness in harsh environmental conditions, where frames may vary significantly, enabling the model to identify vehicle activities independently of the frame's background. However, a notable limitation is that this approach does not account for the temporal sequence of activities, thereby ignoring the interactive dynamics of earthmoving operations and their impact on productivity. To address this, a sequential learning approach has been employed, leveraging bounding boxes to calculate distances between the equipment and assigning sequential labels to

video frames. The sequential learning enables the model to capture the temporal relationships between activities, offering a more comprehensive analysis of earthmoving operations and productivity.

Sequential Learning

The data preparation was mostly identical to the previous section. However, for sequential learning, the sliding window approach has been used instead of MSSS. Where the sliding window is set to a fixed sequence length = 25, and a step size = 10. This approach creates sequences of frames, where each sequence represents 25 frames of video data. The window starts at the beginning of the video and creates a sequence, then slides forward 10 frames to create the next sequence, and so on. The frame data is represented as a list, $F_1, F_2, F_3, \ldots, F_n$ where n is the total number of frames. A sliding window of length L (here L = 25) with a step size S (here S = 10) generates subsequences of frames:

$$S_1 = [F_1, F_1, .., F_{25}],\ S_2 = [F_{11}, F_{12}, .., F_{35}] \quad (5)$$

The sliding window process is applied to each video, where the activities, equipment, and bounding boxes are combined to create a sequence for each window. All the activities are converted to a multi-label format to encode the multi-label sequential activity. Each activity is represented sequentially, corresponding to possible actions (Idle = 0, Swing Bucket = 1, Load Bucket = 2, Dump = 3, Move = 4). For each frame i, the mapping is stored as $M_{f,i}$= {activities: (e, a_e, i), bounding boxes: B_i, productivity: p_i }, where $p_i \in R$ is productivity by assigning the weights for each working mode, a = classes of activities, and e = type of equipment. The final output was serialised into a JSON file.

Figure 5: Methodology for sequential learning used for ActDNN.

As shown in Figure 5, the architecture of the developed sequential learning model aims to effectively process these structured sequences of video frames and their associated bounding box data. This machine learning model integrates a Convolutional Neural Network (CNN) and a transformer encoder with self-attention to predict activity labels and productivity scores for multiple pieces of equipment over sequences of video frames. The CNN backbone based on ResNet-18 is used to extract spatial features, while the transformer encoder with self-attention is used to model temporal relationships. ResNet-18 is pretrained on ImageNet, providing a robust initialisation for spatial feature extraction from 8 video sequences.

As shown in Figure 5, the fully connected layer of ResNet-18 is replaced with a custom linear layer that outputs a feature vector of size cnn_out_dim = 512. Each image from 8 video sequences is passed through this CNN, transforming the input of shape (B×T, Ch, H, W), where B is the batch size, T is the sequence length, Ch is the number of channels, and H and W are the height and width of the frames, into a feature vector of shape (B, T, cnn_out_dim). These sequential feature vectors are then processed by a transformer encoder to capture temporal dependencies. The transformer encoder consists of 4 layers, each containing a multihead self-attention mechanism and a position-wise feed-forward network. The attention mechanism uses n_{heads}=8 attention heads, allowing the model to attend to the temporal relationships simultaneously from the 8 video sequences. The input of the transformer is projected to a dimensionality of transformer_dim = 512 to match the output dimensions of the CNN. The transformer updates these feature representations by computing the self-attention scores Attention $(Q, K, V) = SoftMax\left(QK^T/\sqrt{d}\right)V$, where Q, K, V are query, key, and value matrices derived from the input features, and d is the feature dimension. The output of the transformer is shaped as (B, T, transformer_dim), used for two purposes:

Sequential activity classification, each equipment type is associated with its own fully connected layer. These layers map the transformer outputs to activity probabilities for each equipment, resulting in predictions with a shape of B, T, num_equipment, and num_classes. Cross-entropy loss is used to compute the classification loss defined as:

$$L_{activity} = -\sum_{i=1}^{B}\sum_{t=1}^{T}\sum_{e=1}^{E}\sum_{c=1}^{C} y_{i,t,e,c} \log\left(\hat{y}_{i,t,e,c}\right) \quad (6)$$

where E is the number of equipment, C is the number of classes, and y is the actual classes, $\hat{y}$ and is the predicted classes, i = batch instance, t = time step, e = equipment type and c = activity class.

Productivity prediction, the mean transformer output across the sequence length T is passed through another fully connected layer to produce a single output per batch, yielding predictions of shape (B, 1). The regression loss is computed using mean squared error (MSE), defined as

$$L_{productivity} = -1/B \sum_{i=1}^{B} (\hat{p}_i - p_i)^2,$$

where p_i is the actual productivity and $\hat{p}_i$is the predicted productivity. During training, the Adam optimiser is used with a learning rate of $\alpha = 1 \times 10^{-4}$. Finally, the total loss is a weighted combination of the activity

classification loss and the productivity regression loss. Training is performed over 20 epochs with a batch size of 16, as shown in Figure 6.

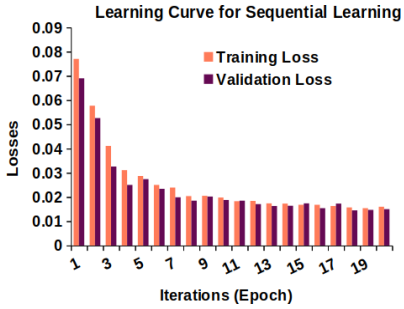

Figure 6: Learning curve for Sequential ActDNN

This learning curve shows the comparative analysis of the training loss (orange bars) and validation loss (purple bars). During epoch 1, the training and validation loss started from 0.078 to 0.07, decreasing exponentially until 9 epochs and getting saturated. As both losses align closely, the algorithm can generalise well to unseen data. The consistent convergence and stabilisation of losses after epoch 13 demonstrates that the model has reached a stable state and training and validation datasets. To prevent excessive computational loss, early stopping was employed when the model ceased to improve, making further training unnecessary. The trained algorithm has been tested on the unseen video sequence (134516), which consists of the combination of two trucks and one excavator, and the result has been plotted on temporal distribution with a 97.79%, 89.67% and 86.48% sequential activity prediction accuracy for excavator, $truck_1$ and $truck_2$ as shown in Figure 7.

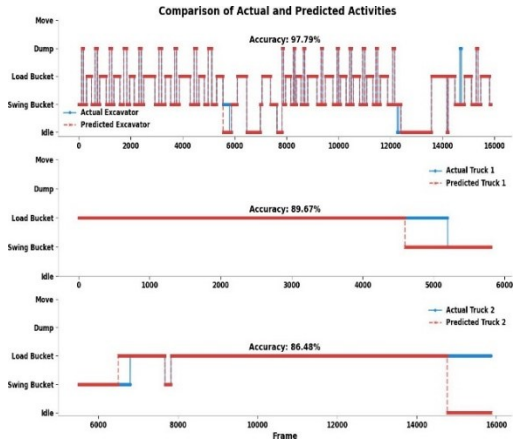

Figure 7: Comparison of Sequential Activity prediction using ActDNN.

Each panel represents the activity sequence of a specific piece of equipment, where the y-axis represents activity types (Idle, Swing Bucket, Load Bucket, Dump, Move), and the x-axis denotes the frame indices in temporal order. The red dashed line represents the predicted activity, while the blue solid line indicates the actual activities. The excavator subplot shows frequent transitions between activities, such as swing bucket, dump, and idle. The algorithm demonstrates 97.79% accuracy for the excavator, with minor deviations observed during activity transitions. The second subplot (for $Truck_1$) reveals fewer transitions, with the load bucket dominating the sequence. The activity recognition reaches an accuracy of 89.67%, with some deviations occurring towards the swing bucket.

In the case of $Truck_2$, the activity sequence is relatively simple, with load bucket dominating throughout the time. However, the algorithm is prone to errors during transition periods between swing bucket and load bucket, reducing the accuracy to 86.48%. The primary reason for prediction errors is due to the temporal overlaps and classification uncertainties occurring. Figure 8 represents the absolute predicted productivity absolute loss across sequential indices. Where the x-axis shows the sequence indices of the video frames, corresponding to time, while the y-axis shows the absolute loss values, ranging from 0 to approximately 0.3. The loss values represent the difference between the predicted productivity and the actual productivity values for each sequence in the video. A higher loss indicates that the model's predictions are farther from the actual values, suggesting poorer performance. The goal is to minimise this loss during training so that the model can make more accurate predictions for productivity in future sequences.

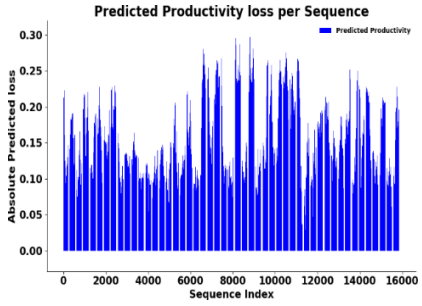

Figure 8: Sequential productivity loss

The closely spaced blue bars reflect the fine-grained variations in predicted loss across a large number of sequences. The fluctuations, with periodic clusters of sequences exhibiting higher loss values, followed by intervals of lower loss. This variability shows that the model performs inconsistently across different sequences, which is due to the changes in operational conditions, e.g. (repetitive activities (Swing Bucket, Load Bucket, Dump). The loss rarely exceeds 0.3, indicating that the model maintains good accuracy for most sequences, but the high-loss sequences occur for idle states. Lastly, Table 2 shows the comparative studies for activity classification of heavy equipment on construction sites.

Table 2: Comparative studies used DL for heavy equipment classification and detection on construction sites.

References	DL Method	Accuracy	No of Classes
Shen et al. (2024)	Temporal Convolutional Neural Network (TCN)	0.945	8
Arabi et al. (2020)	Convolutional Neural Network (CCN)	0.900	6
Lu et al. (2021)	Residual Neural Network (ResNet) CNN	0.952	11
Slaton et al. (2020)	CNN, Long short-term Memory (LSTM)	0.744	6
Akhavian & Behzadan (2015)	Artificial Neural Network (ANN), K-nearest neighbour (KNN), Decision Tree (DT)	0.985	5
This Study	ResNet-CNN	0.9982	5
This Study	ResNet-CNN, Transformer	E = 98%, T1 = 96% and T2 = 86%	5
This Study	ResNet-CNN, Transformer, Self-attention	Absolute loss = 0.1555	

Discussion and Result analysis

This study addresses two key challenges on construction sites. First, the missing or incomplete data due to often harsh site conditions, which was addressed by employing the independent learning approach. To support that, random image frames from different construction sites were used in the study. The developed algorithm has shown 99.82% accuracy in predicting multiple equipment activities, independent of the background variations across job sites. This algorithm was tested on an unseen dataset to ensure its generalization. These results suggest that the algorithm can be used in wider scenarios, as it represents a certain degree of 'zero-shot' capabilities in multiple scenarios having one excavator and up to three trucks. The performance of the algorithm varies based on the number of pieces of equipment embedded in the scenarios, 99.82% for the excavator, 99.90% for $Truck_1$ and excavator, 100% for $Truck_1$, $Truck_2$, $Truck_3$ and excavator, and 99.94% for $Truck_2$ and excavator. This demonstrates that while the algorithm is effective across various scenarios, accuracy can be influenced by the number of equipment types being tracked simultaneously.

The second practical problem is that in large-scale construction projects, delays often occur due to a lack of proper daily monitoring, leading to significant losses in terms of time and resources. To address this problem, an upgraded version of the algorithm incorporates temporal information into the previous framework. This upgrade enables the system to perform more efficiently by leveraging sequential information to improve predictions over time. The upgraded algorithm integrates the attention-based approach with the transformer, designed to reveal the temporal activities. In the studied case, the algorithm selectively concentrates on the type of equipment and the operation being performed. By doing this, it ensures that the model can better track the time-sensitive activity sequences, with 97.79%, 89.67%, and 86.48% prediction accuracy for the excavator, $truck_1$, and $truck_2$, respectively, with a productivity prediction absolute loss of 0.1555%. This advanced version of the algorithm offers a robust multitasking solution capable of effectively handling the temporal complexities of construction site operations.

Conclusion

Finally, this study addresses two major challenges in construction sites. First, dealing with unstructured image frames where data leakage and missing data are common is tackled by ActDNN, which achieves notable accuracy of 99.82% in predicting multiple equipment activities, regardless of background variations. This algorithm has been rigorously tested on an unseen dataset to validate its robustness, with its performance varying based on the number of pieces of equipment involved. The second challenge is the lack of effective daily monitoring, which is mitigated by an advanced version of the algorithm that incorporates temporal information. Leveraging an attention-based approach with a transformer, this algorithm focuses on the type of equipment and the operation being performed, achieving prediction accuracies of 97.79% for the excavator, 89.67% for $truck_1$, and 86.48% for $truck_2$. Additionally, the absolute loss in productivity prediction is approximately 0.1555%. This approach provides an intelligent multitasking solution capable of addressing the temporal complexities of construction site operations. Together, these algorithms form a robust framework for automated construction equipment monitoring and productivity analysis. They can identify activities and equipment from random images while enhancing the applicability of vision-based methods for sequential activity and productivity analysis. These advantages make the proposed solution a scalable and potentially commercially viable approach to automating construction site operations.

Acknowledgments

This document was derived from the author guidelines used for the 2003, 2005, 2008, 2014, and 2017 Building Simulation Conferences and the 2019 EC3 Conference.

References

Akhavian, R. and Behzadan, A. H. (2015). Construction equipment activity recognition for simulation in-put modeling using mobile sensors and machine learning classifiers. Advanced Engineering Informatics, 29(4):867–877.

Arabi, S., Haghighat, A., and Sharma, A. (2020). A deep-learning-based computer vision solution for construction vehicle detection. Computer-Aided Civil and Infrastructure Engineering, 35(7):753–767.

Arditi, D. and Mochtar, K. (2000). Trends in productivity improvement in the us construction industry. Construction Management & Economics, 18(1):15–27.

Behzadan, A. H., Aziz, Z., Anumba, C. J., and Kamat, V. R. (2008). Ubiquitous location tracking for context-specific information delivery on construction sites. Automation in construction, 17(6):737–748.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

Kamat, V. R., Martinez, J. C., Fischer, M., Golparvar-Fard, M., Peña-Mora, F., and Savarese, S. (2011). Research in visualization techniques for field construction. Journal of construction engineering and management, 137(10):853–862.

Kassem, M., Mahamedi, E., Rogage, K., Duffy, K., & Huntingdon, J. (2021). Measuring and benchmarking the productivity of excavators in infrastructure projects: A deep neural network approach. Automation in Construction, 124, 103532.

Kim, H., Ham, Y., Kim, W., Park, S., and Kim, H. (2019). Vision-based nonintrusive context documentation for earthmoving productivity simulation. Automation in Construction, 102:135–147.

Kim, J. and Chi, S. (2020). Multi-camera vision-based productivity monitoring of earthmoving operations. Automation in Construction, 112:103121.

Kim, J., Chi, S., and Hwang, B.-G. (2017). Vision-based activity analysis framework considering interactive operation of construction equipment. In Computing in civil engineering 2017, pages 162–170.

Kim, J., Chi, S., and Kwon, T. (2016). Construction entities tracking based on functional integration and online learning with site-customized datasets. In Proceedings of the CIB World Building Congress, pages 1118–1128.

Kim, J., Ham, Y., Chung, Y., and Chi, S. (2018). Camera placement optimization for vision-based monitoring on construction sites. In Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC).

Korea Construction Technology Promotion Act, E. d. a. and 99, s. o. t. R. o. K. (2017). http://law.go.kr/ /, (2016. In Computing in Civil Engineering 2017, page Accessed date: 9 Dec 2024.

Lu, J., Yao, Z., Bi, Q., and Li, X. (2021). A neural network–based approach for fill factor estimation and bucket detection on construction vehicles. Computer-Aided Civil and Infrastructure Engineering, 36(12):1600–1618.

Merrillees, M. and Du, L. (2021). Stratified sampling for extreme multi-label data. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 334–345. Springer.

Roberts, D. and Golparvar-Fard, M. (2019). End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level. Automation in Construction, 105:102811.

Rogage, K., Mahamedi, E., Brilakis, I., & Kassem, M. (2022). Beyond digital shadows: A Digital Twin for monitoring earthwork operation in large infrastructure projects. AI in Civil Engineering, 1(1), 7.

Saif, W., & Alshibani, A. (2024). A close-range photogrammetric model for tracking and performance-based forecasting earthmoving operations. *Construction Innovation*, *24*(1), 164-195.

Saif, W., Rogage, K., Martinez, P., & Kassem, M. (2025). Decarbonising construction equipment: Management practices and strategies for net zero in UK infrastructure projects. *Building and Environment*, *270*, 112503.

Shen, Y., Wang, J., Feng, C., and Wang, Q. (2024). Dual attention-based deep learning for construction equipment activity recognition considering transition activities and imbalanced dataset. Automation in Construction, 160:105300.

Slaton, T., Hernandez, C., and Akhavian, R. (2020). Construction activity recognition with convolutional recurrent networks. Automation in Construction, 113:103138.

Yang, Z., Yuan, Y., Zhang, M., Zhao, X., and Tian, B. (2019). Assessment of construction workers' labor intensity based on wearable smartphone system. Journal of construction engineering and management, 145(7):04019039.