PRODUCTIVITY MONITORING OF CONSTRUCTION WORKERS BASED ON SPATIOTEMPORAL ACTIVITY RECOGNITION

Ghazaleh Torabi

A Thesis in

The Department of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirement

For the Degree of

Master of Applied Science (in Electrical and Computer Engineering) at

Concordia University

Montreal, Quebec, Canada

March 2022

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Ghazaleh Torabi

Entitled: PRODUCTIVITY MONITORING OF CONSTRUCTION WORKERS BASED ON SPATIOTEMPORAL ACTIVITY RECOGNITION

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Krzysztof Skonieczny — Chair

Dr. Yiming Xiao (CSSE) — Examiner

Dr. Amin Hammad — Supervisor

Dr. Nizar Bouguila — Supervisor

Approved by

Dr. Yousef Shayan, Chair
Department of Electrical and Computer Engineering

Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Date

ABSTRACT

Productivity Monitoring of Construction Workers Based on Spatiotemporal Activity Recognition

Ghazaleh Torabi

Workers' productivity monitoring is an essential but time-consuming part of large construction projects. Therefore, automating this process using surveillance cameras has gained a lot of interest among researchers. A human observer can extract both detailed and abstract information from surveillance videos to estimate productivity. Humans first gain a high-level understanding of the scene and then pay attention to low-level details. Automating this process requires computers to understand videos at different levels as well. However, previous studies only focused on low-level activities. In addition, the three-stage activity recognition method adopted by previous studies consists of separately optimized worker detection, tracking, and activity classification modules. The three-stage method propagates errors through its modules, does not leverage the scene context, and was trained and tested on trimmed datasets in the previous studies. To address these limitations and research gaps, this thesis aims to: (1) propose a fully optimized method for activity recognition of construction workers in untrimmed surveillance videos, (2) use a combination of workers' low-level activities to understand their higher level micro-tasks, (3) calculate the percentage of workers' time spent on different activities and micro-tasks, (4) identify low productivity and its underlying reasons by calculating the percentage of activities for each micro-task, (5) identify idling and its underlying reasons, and (6) combine resource monitoring with progress monitoring by recognizing built construction elements, calculating their completion time, the average number of utilized workers, and the percentage of their time spent on each related micro-task. The proposed fully optimized activity recognition method improved the activity classification accuracy of the three-stage method by 15%, proving that a fully optimized method is superior to the previous separately optimized methods. The proposed productivity monitoring framework was applied to a two-hour video of workers assembling footing formwork, and a six-hour video of footing formwork assembly and installation of footing reinforcement bars, showing that the framework is promising. A detailed analysis is conducted on underlying reasons for low productivity and idling, which proved that activities alone are not informative enough for decision making.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Dr. Amin Hammad and Dr. Nizar Bouguila, who provided me with the opportunity to learn from them and supported me throughout my masters. I appreciate their time, contributions, and patience. Their guidance and attitude towards the hardships and uncertainties of research work have greatly shaped me and will continue to stay with me.

I would also like to thank the members of my defense committee, Dr. Skonieczny and Dr. Xiao for spending their precious time reading this thesis and providing me with their valuable feedback.

I was very fortunate to meet my fellow lab members. Although we were not able to meet in person for most of my program due to COVID lockdowns, they were still very warm and supportive during my two years at Concordia.

Lastly, I would like to express my deep gratitude to my father for his love and support in all stages of my life. I would not have achieved this without him.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
ACAR	Actor-Context-Actor Relation
ACT	Action Tubelet Detector
AP	Average Precision
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CV	Computer Vision
DL	Deep Learning
FLOPs	Floating Point Operations
FOV	Field of View
FP	False Positive
FPS	Frame Per Second
GPU	Graphical Processing Unit
HMM	Hidden Markov Model
IoU	Intersection over Union
LRCN	Long-term Recurrent Convolutional Network
LSTM	Long Short-Term Memory
MPDM	Method Productivity Delay Model
NMS	Non-Maximum Suppression

OF	Optical Flow
R-CNN	Region-based Convolutional Neural Network
RGB	Red, Green, and Blue
RNN	Recurrent Neural Network
SORT	Simple Online and Real-Time
SSD	Single Shot Detector
STEP	Spatio-Temporal Progressive
SVM	Support Vector Machine
TACNet	Transition-Aware Context Network
TP	True Positive
TSN	Temporal Segment Network
YOLO	You Only Look Once
YOWO	You Only Watch Once

Chapter 1: Introduction

1.1 Background

"Economists have been saying it, so have constructors, organized labor - everybody: to remain competitive, we have to produce more for each dollar spent on construction" [1]. Although construction is one of the largest industries around the world, its productivity is much lower than the other large industries. Labor productivity rate is one of the most important factors for contractors in determining the labor cost and the success or failure of the project. Therefore, companies should maintain an acceptable productivity rate by providing managers with accurate and timely data. Managers use these data to determine if their management effort is effective, detect trends and make corrective decisions, determine the effect of recent imposed methods or conditions, identify underlying reasons behind low and high productivity, and compare different projects [2]. Despite its importance, labor productivity is usually estimated inaccurately [2].

"The productivity of a process can be measured indirectly by observing the activity of its resources" [1] (e.g., workers). Most commonly used productivity measurement methods require some form of data collection that is traditionally done through manual supervision and tracking (either in person or from videos) of resources (i.e., workers and equipment), task progress, and activities, as well as interviews, surveys, and interactions with workers. These approaches are labor-intensive, time-consuming, and may not be detailed or precise enough, especially for large projects. This results in inaccurate productivity estimation, ineffective management, and consequently low productivity of construction operations compared to other industries. Therefore, there is a clear need to improve the speed, accuracy, and efficiency of productivity measuring methods through automation. Since surveillance cameras are already installed on most construction sites nowadays, computer vision can replace humans in collecting some of the required data for productivity measurement and analysis.

1.2 Problem Statement and Research Gaps

A construction operation consists of different levels of work, ranging from activities (achieved in few seconds) to micro-tasks (achieved in few minutes), to tasks commonly used in construction schedules (achieved in weeks or months); each resulting in the realization of different parts of the

final construction products. Productivity measurement and analysis require collecting data from all above levels.

Several research works have leveraged computer vision (CV) to automatically recognize resource activities [3–8], which can be used to understand how resources are utilized and calculate the duration of value-adding and non-value-adding activities. Figure 1-1 shows the most recent method (referred to as the three-stage method in this thesis) that has been widely adopted in multiple studies such as [3, 6, 7]. The three-stage method consists of three main modules: (1) detection, (2) tracking, and (3) activity classification, which are optimized separately and are applied sequentially on the input video. While this method has been showing promising results in research, it is not applicable to real-life scenarios due to the practical limitations explained below that reduce its performance.

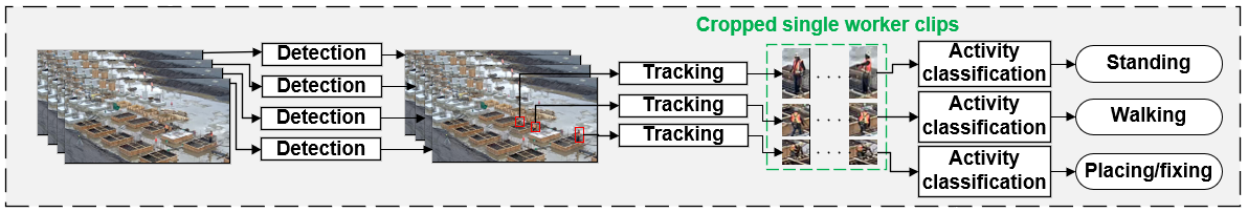


Figure 1-1 The inference workflow of the three-stage method

Productivity analysis requires reasoning over long durations. Therefore, the input videos are usually around 8-10 hours in length. These videos are different than some publicly available activity recognition video datasets known as trimmed datasets [9–11], which are carefully handcrafted to include only one activity in each video clip. Trimmed datasets require only a single label per input (i.e., few seconds video clip or multiple consecutive frames) for training, and the trained model outputs a single label per input during inference as well. In contrast to these datasets, there are untrimmed datasets [12–14] that can contain some unrelated frames or even multiple consecutive activities in each video clip. Untrimmed datasets are easier to gather with less manual effort allowing the creation of large datasets. However, activity recognition for these datasets is generally more challenging and shows lower performance [12, 13]. Some of these datasets are shown in Table 2-1 and will be briefly reviewed in Section 2.3. The long videos coming from surveillance cameras of construction sites are from the second type (i.e., untrimmed), with workers frequently switching between activities. These videos either require (label, temporal boundary) pairs or per-frame labels for training. Similarly, the trained model should output the same type of

data. However, the way that the three-stage method has been applied in the previous studies is more suitable for trimmed video clips. Therefore, it is expected that the performance of this method drops when applied on long untrimmed videos [15], such as the ones needed for productivity analysis of construction workers. Previous studies [3, 5–7] suggested using short input video clips to lower the chance of having untrimmed video clips. However, this does not guarantee that all the frames of a video clip are focusing on a single activity and do not contain, for example, the ending or the beginning of another. In order to get a good performance on untrimmed videos, the model should be both trained and evaluated on untrimmed videos [15]. Therefore, this thesis suggests using an alternative approach of per-frame and per-worker annotation and using untrimmed video clips for both training and evaluation of the model. Per-frame activity recognition for every worker allows obtaining the temporal boundaries of activities, improves the performance of the model near these boundaries, and gives more fine-grained, detailed, and accurate results on untrimmed videos [12, 16–18].

Another shortcoming of the three-stage method is that each module is optimized separately. This approach does not guarantee the optimization of the entire method and raises several issues. First, the output of the object detection module in the three-stage method is a bounding box around each worker. When testing object detection models, an IoU threshold is usually specified. If the detected box has an IoU higher than this threshold with the ground truth, then the detection is considered as a true positive [19, 20]. A typically acceptable threshold in object detection challenges is 0.5, while a relatively high threshold is 0.75 [19, 20]. IoU of 1.0 is hardly expected from these models. However, for the activity classification module (the third module) to display its best performance, the detected workers must be entirely captured in the input video clip of this module. Second, some level of error is expected from both the object detection and tracking modules, which makes it even harder to satisfy the ideal condition for the activity classification module and propagates the error. Third, the spatial information around workers (such as objects and tools) can help with activity classification, while this information can be removed by the object detection module that only focuses on detecting workers. Additional modules such as Conditional Random Field (CRF) used in [7] may be needed to incorporate the additional spatial information. It is worth mentioning that some background information from the areas around workers will be added to the video clip before feeding it to the activity classification module of the three-stage method. This is because the input video of the activity classification module is cropped around the smallest box that covers

all tracked bounding boxes of each actor (shown in Figure 1-2, and referred to as the extended bounding box in [5]). However, the size of the extended bounding box depends on the movement range of the workers and does not guarantee to eliminate the issue entirely. One possible way to solve this issue is by further extending the extended bounding box before feeding it to the activity classification module, but depending on the movement range of the workers, this increases the possibility of including another worker in the video clip, which is against the necessary condition of the input of the activity classification module. The activity classification module in this method requires its input video to contain only a single worker.

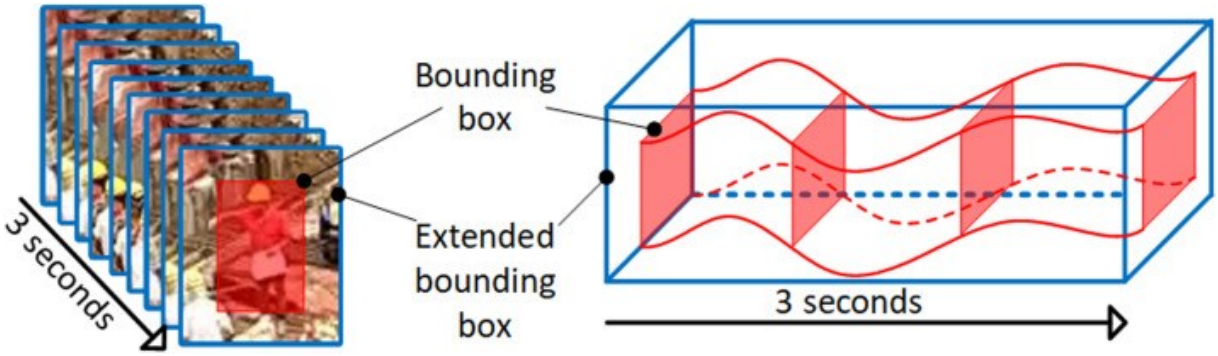


Figure 1-2 Extended bounding box [5]

In addition to limitations of the previous activity recognition methods, activities that have been recognized in previous studies [4, 5, 7] (e.g., walking, hammering, transporting) are very short in duration, and not always informative enough about the type of construction task that is being done, since many of them exist as part of several different construction tasks. As a result, these activities are not comparable to the daily schedule and cannot be directly used for identifying low productivity among workers and making constructive decisions to improve the situation. It is the combination or the sequence of such activities (referred to as micro-tasks in this study) that can be further linked with the tasks in the daily schedule, which can provide managers with useful high-level information. Although not informative alone, activities show whether workers are being productive enough for the recognized micro-tasks, and what can be done to improve productivity based on the nature of the recognized micro-tasks. For example, long durations of walking may be expected to a certain extent for some micro-tasks, or can be a sign of low productivity for others.

Micro-task recognition for workers is very challenging. Most heavy construction equipment (e.g., cranes, excavators) have production cycles. Their activities (e.g., digging, swinging, dumping) ideally follow a clear sequence, which can be defined in discrete-event simulation tools such as

Cyclone [21] or Stroboscope [22]. They have a limited range of movements and usually do not perform unexpected activities. Unless they are idling, the activities continuously follow one another, forming predefined cycles that can be used to recognize their micro-tasks. The duration of these cycles and the activities durations within these cycles can be used as input for productivity calculation [23, 24]. On the other hand, construction workers do not have strict cyclic activities. They have a large range of movements with most of them being more complex compared to those of construction equipment. They do not always follow the same sequence of activities for each micro-task and often violate the expected sequence. In addition, workers cannot be expected to work continuously; they regularly pause to take a break, check their work, or discuss it with their colleagues, especially if they are working in groups. This makes the recognition of micro-tasks and idling more challenging for workers compared to equipment.

In addition, productivity is measured as the relationship between the input (time consumed by resources such as workers and equipment) and the output (built products) of an operation. This requires monitoring both the resources (e.g., through micro-task recognition and activity recognition) and the site progress. Although this has been addressed in multiple studies for excavators, there is not much research on how to combine resource monitoring and progress monitoring for the automatic extraction of workers' productivity data. Combining the extracted information from all the above levels of an operation (i.e., activities, micro-tasks, and built products) allows the detailed analysis of the productivity of workers involved in tasks at different parts of the site and helps managers identify issues and their underlying causes, and make informed decisions to improve the conditions.

1.3 Research Objectives

At a high level, this study aims to fill the research gaps in automatic productivity monitoring of construction workers by: (1) recognizing workers' activities and the percentage of their time spent on each, (2) recognizing workers' micro-tasks and the percentage of their time spent on each (3), identifying low productivity and its underlying reasons by calculating the percentage of workers' times spent on different activities for each micro-task, (4) identifying idling and its underlying reasons, and (5) combining resource monitoring with progress monitoring by recognizing the built products, calculating their completion times, the average number of utilized resources (i.e., workers), and their activities and micro-tasks.

In order to realize the first high-level objective (i.e., activity recognition), the secondary objectives of: (1) proposing a fully optimized activity recognition method that does not have the limitations of the previous work (i.e., the three-stage method) for untrimmed videos, and (2) verifying the superiority of the proposed method over the previous three-stage method for the task of construction workers' activity recognition should first be realized.

1.4 Thesis Organization

Figure 1-3 shows an overview of the proposed framework. Chapter 3 is focused on detecting construction workers and classifying their activities. In Chapter 4, activities are turned into a higher level and more meaningful micro-tasks, that can be further used in the future to recognize high-level construction tasks that are comparable to the daily schedule. Furthermore, given that productivity is defined as the relation between input resources (e.g., number and work-hour of workers) and outputs (e.g., number of completed products), an object detection method is used to measure the productivity of the site by detecting completed products and computing their completion times. Progress monitoring is combined with resource monitoring (i.e., activity and micro-task recognition), to obtain a full productivity monitoring framework by taking both input and output into account. Six main productivity data are extracted in the proposed framework; they are the number of workers and their location, activities and their duration, micro-tasks and their duration, the number and location of completed products, the completion times of products, and the average number of utilized resources for each. The collected data can be combined in the future to measure productivity in a quantitative way using available methods such as work sampling, which will be explained later in Section 2.2

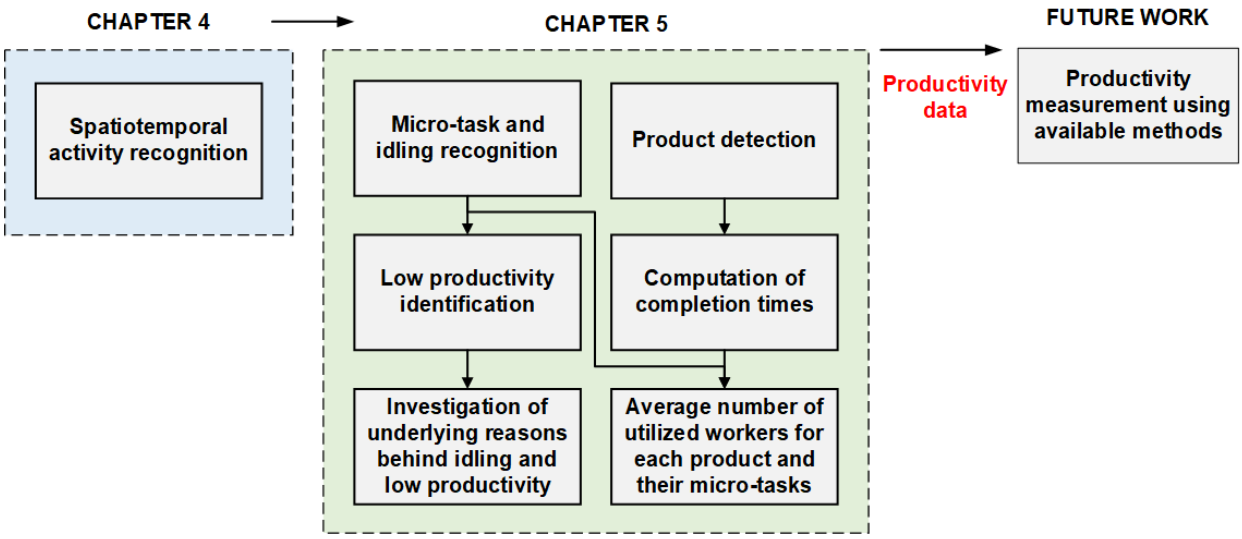


Figure 1-3 Overview of the proposed framework

The structure of this thesis is as follows:

Chapter 2 - Literature Review: This chapter reviews the literature about commonly used productivity measurement methods, CV-based activity recognition datasets and methods in the computer science domain, and productivity monitoring in the construction domain for workers.

Chapter 3 – CV-based Spatiotemporal Activity Recognition: This chapter is focused on the first module of the productivity monitoring framework. It presents the proposed spatiotemporal activity recognition method (i.e., YOWO) and its improved version (YOWO53), and compares them with one of the previously widely used methods in the construction domain (i.e., three-stage method).

Chapter 4 – CV-based Micro-task Recognition and Productivity Monitoring: This chapter first presents the overall productivity monitoring framework and two of its modules (micro-task and idling recognition, and product detection) in detail. It then proceeds with discussing how to use the framework to detect low productivity, and identify reasons behind low productivity and idling through two detailed case studies.

Chapter 5 – Summary, Contributions, and Future Works: This chapter presents a summary of the thesis, its contributions, and limitations as well as future research directions.

Chapter 2: Literature Review

2.1 Introduction

Productivity analysis requires relating the input (e.g., resources such as workers, equipment, or materials) of an operation to its output (i.e., built products). Researchers have studied the role of CV in the automatic monitoring of both ends of construction operations (i.e., input and output). For example, monitoring the input requires understanding the type and number of resources on site, their locations, and how the key resources are utilized. The recent CV-based solutions that have been proposed to extract these data use different combinations of object detection and localization, object tracking, and activity recognition [4–8, 25–30]. CV-based monitoring of the output (i.e., progress monitoring) has been studied extensively as well. Researchers have tried different ideas such as scanning the as-built 3D point clouds [31], material classification [32, 33], and comparison of the site with the as-planned 4D BIM simulation [34–37].

With the rise of advanced deep learning (DL) methods, DL-based activity recognition has gained a lot of interest among construction researchers for resource monitoring, as it can provide information about how each resource is being utilized. However, unlike state-of-the-art object detection methods, their application in the construction domain is still an ongoing research.

Activity recognition methods can be classified based on their input or output types. Input video clips can be trimmed or untrimmed, while the outputs depend on the task at hand. As was explained in Section 1.2, trimmed video clips are temporally trimmed around single activities, while untrimmed video clips can contain some unrelated frames or multiple consecutive activities. In addition to the input, there are several different outputs based on the task at hand. Activity classification is one of these tasks and aims to choose a single label from a set of predefined labels for the input video clip, whether it is trimmed or untrimmed [13]. This method is the ideal approach when the input video clip contains a single actor and a single activity. The construction community has been using this approach to classify the activities in construction site videos so far [3, 5–7]. However, since there are multiple workers in the site videos, the first step is to isolate the workers using available methods and generate cropped video clips for each (Figure 1-1).

Spatiotemporal activity recognition is another task and is useful when multiple actors are performing different activities in the videos. Spatiotemporal activity recognition methods produce

bounding boxes around workers as well as their activity labels [38–42]. Considering the nature of construction site videos, spatiotemporal activity recognition is the most suitable approach.

The rest of this chapter contains a brief review of common labor productivity measurement methods, some commonly used benchmark trimmed and untrimmed video datasets for human activity recognition, CV-based human activity recognition methods for both activity classification and spatiotemporal activity recognition tasks, and available CV-based workers' productivity monitoring studies in the construction domain. In addition, the limitations and research gaps in the previous CV-based workers' activity recognition and productivity monitoring studies are discussed in this chapter. This review is focused on more recent convolutional neural network (CNN) based methods as they proved to be more promising than their traditional counterparts. The identified limitations motivated the proposed methods of this research in Chapters 3 and 4.

2.2 Productivity Measurement Methods

Productivity has various definitions. One of the common definitions of labor productivity, which is also adopted in this study, is the quantity of output, produced by a given quantity of labor input (i.e., output per man-hour) [43] shown in Equation (1). The inverse of this value shown in Equation (2), known as unit rate [44], is also commonly used.

$$labor\ productivity = \frac{Quantity\ of\ output}{Work\ Hours} \quad (1)$$

$$labor\ productivity = \frac{Work\ Hours}{Quantity\ of\ output} \quad (2)$$

The two above definitions measure productivity accurately. However, easier and indirect ways are also used for productivity measurement in practice. Some of the widely used techniques are briefly described below.

(1) Field rating: This technique uses the activity level of the site as an indirect way of estimating productivity. In order to do this, the foreman makes random observations of workers and categorizes them into "working" and "non-working" categories. Field rating is then calculated as the percentage of "working" instances to the total number of observations, plus 10% for foreman's and supervisor's work. Field rating does not give any information about low productivity reasons [1].

(2) Work Sampling: Work Sampling is an indirect measurement technique for labor productivity. It is based on workers' time utilization in three categories of Direct Work (productive), Support Work (semi-productive), and Delays (non-productive). Any activity that directly adds value is considered as Direct Work (e.g., hammering, placing/fixing rebars). Activities that do not directly add value but assist in value-adding activities are considered as Support Work (e.g., transporting). Delays are activities that waste time (e.g., idling). The method requires random selection of a group of workers, monitoring them for a specific period, and taking note of their Direct Work, Support Work, and Delays in fixed time intervals (e.g., 10-minute intervals). The percentage of the time spent in each of the three categories is an indirect indication of workers' productivity performance. However, this method only measures how resources (i.e., workers and their time) are utilized, and does not consider the output [1].

(3) Five-Minute Ratings: This method requires identifying multiple members of the crew and observing them for a specific period of time in 5-minute intervals. For each crew, if they are active for half of the interval, the entire interval is noted as positive. The number of positive observations to the total number of observations is considered as a measure of effectiveness, and indirectly as a measure of productivity [1].

(4) Craftsman Questionnaire: This technique provides craftsmen with a questionnaire to collect important productivity-related data, such as material availability, site layout, equipment and tool availability, and lost hours caused by these issues. Using the filled questionnaire, managers can identify wasted time and its reason [1].

(5) Forman delay survey: Forman delay survey is used to collect the number of lost hours due to delays and rework. If a specific delay or rework cause is identified to waste too much time, required actions are taken to solve the issue [1].

(6) Method Productivity Delay Model (MPDM): MPDM is a direct productivity measurement method. First, the production unit, the production cycle, the leading resource, and possible types of delays are identified. The production unit is a measurable amount of work that can easily be identified visually. The production cycle is the time it takes workers to place one production unit. The leading resource is the most fundamental resource of the operation. Next, an observer measures the cycle duration of each production unit, and identifies delays during the cycles. This information is then used to calculate the productivity rate and identify the

reasons behind low productivity through a series of computations. The applicability of this method is limited if the cycle time is too short or too long to keep track of [1].

(7) Crew Balance Charts: Crew balance charts are appropriate for cyclical tasks. A bar chart is created in this method for the task at hand, with each bar representing an involved worker or equipment. Each bar is made up of the duration of all the activities (whether productive, non-productive, or idling) that are performed by the associated worker or equipment during the cycle time. The chart can be used to compare interrelationships between workers and equipment and identify inefficiencies [1].

(8) Simulation: simulation-based methods (e.g., Cyclone) model an operation using its involved activities, their logical relationships, durations, and resources. After modeling an operation, the simulation can be used to compute the productivity, duration, and cost of the operation [1].

All above methods require input data, such as the crew size, their value-adding, non-value-adding, and idling activities and their durations, the type of the task and its duration, cycle times, number of completed products, and product completion times. These data are only obtainable from the construction site itself. This requires either in-person or remote (e.g., through recorded videos) monitoring of the site by supervisors and foremen. Collecting these data is labor-intensive and time-consuming. As a result, it is not done frequently and accurately enough to be used for effective productivity measurement and decision making. Therefore, this research work tries to automate the collection process of some of the above data to facilitate labor productivity measurement.

2.3 Human Activity Recognition Video Datasets

Some of the papers introduced in this section make a distinction between activities and actions. Activities are considered to be made up of simple actions, are generally more complex, and may contain interactions with other people or objects [45]. However, not all papers follow the same definitions. The two terms (i.e. activity and action) are used interchangeably in this thesis when referring to available datasets and previous studies. The dataset prepared for this thesis contains interaction with objects/tools, as well as the interaction between workers (e.g., two workers transporting materials together). However, recognizing workers' interactions is not the focus of this thesis. Additionally, as the activities used in this study and previous studies in construction,

are very short in duration (i.e., few seconds) they are considered to be atomic activities. A summary of different characteristics of some of the publicly available human activity recognition datasets is shown in Table 2-1 and briefly explained below.

AVA: AVA introduced in [12] is an untrimmed video dataset consisting of 430, 15-minute video clips with localized actions in both space and time. These videos are gathered from movies and contain 80 different atomic actions. Videos may contain multiple actors and each actor is labeled separately. Actors may have multiple labels based on their pose, their interaction with other objects, and their interaction with other actors. The dataset contains track annotations as well to track actors in consecutive frames.

Kinetics: The authors of [9] introduced Kinetics-400 in 2017 with the goal of replicating large-scale image datasets, such as ImageNet, for video classification. The dataset contains 400 activity classes from YouTube videos including human-human interaction, and human-object interaction with at least 400 clips in each class. Videos are trimmed and are about 10 seconds. Some videos may contain multiple activities such as texting while driving a car; however, these clips are labeled with only one of these activities. There may be multiple actors in some of the videos, but the videos have a single label as these cases are either human-human interaction, the other actors are performing the same task, or the other actors are not the main focus of the video. They later introduced Kinetics-600 [46] and Kinetics-700 [47] in 2018 and 2019, respectively, as well as the AVA-Kinetics [48] dataset in 2020 which is a combination of AVA and Kintecis-700 with spatial annotations for a single keyframe in each Kintecis-700 video.

Charades: Charades is an untrimmed video dataset from human daily activities that cannot be found in YouTube videos or movies, as opposed to most other large-scale datasets. Annotations include multiple text descriptions, activity labels, activity intervals, and classes of interacting objects. There are 46 objects and 157 activity classes in the dataset and the videos are on average 30 seconds long. Actors may perform multiple activities in sequence, which are labeled along with their respective temporal intervals. Some of the videos contain multiple actors; however, similar to Kinetics dataset, they are not labeled separately [14].

Table 2-1 Human activity recognition datasets and their properties

	Untrimmed			Trimmed		
Dataset	AVA [12]	Charades [14]	ActivityNet [13]	Kinetics [9]	UCF101 [11]	HMDB [10]
Multiple actors	Yes	Yes*	Yes*	Yes*	Yes*	Yes*
Multiple activities per actor	Yes	Yes	Yes	Yes**	Yes**	Yes**
Spatial annotations	Yes	No	No	No	No	No
Temporal annotations	Yes	Yes	Yes	-	-	-
Duration	15m	Avg. 30s	Mostly 5-10m	~10s	Avg. 7.2s	Avg. 3s
Number of videos	430	9,848	27,801	306,245	13,320	6,766
Number of activities	80	157	203	400	101	51
Source of videos	Movies	Crowdsourced daily activities	Online videos	YouTube	YouTube	Movies and Online videos
Year	2018	2016	2015	2017	2012	2011

* There may be multiple people in the clips, but the activities are either human-human interactions, all people are performing the same activity, only one person is the focus of the video, or the other people are not performing any of the predefined activity classes.

** Some clips may contain more than one activity, but they are only annotated under one of the classes.

ActivityNet: ActivityNet is an untrimmed video dataset from human daily activities collected from online sources. The dataset contains video clips of various durations with 203 activity classes. Each untrimmed video may contain multiple activities with their temporal boundaries. ActivityNet does not provide separate labels for each actor similar to Kinetics and Charades [13].

UCF101: UCF101 [11] contains trimmed video clips of 101 activity classes from YouTube. The videos are on average 7.2 seconds and include camera motion and cluttered backgrounds. A subset of activity classes (24 classes) is spatially annotated with bounding boxes around actors. This subset is known as UCF101-24 and can be used for spatiotemporal activity detection tasks. UCF Sports Action dataset is another spatially annotated dataset from 10 classes of various sports.

HMDB: HMDB [10] contains 51 trimmed action classes of 3 seconds in length on average, collected from online sources and movies. The dataset contains additional information on the camera view, visible body parts, camera motion, video quality, and the number of actors involved. JHMDB is a subset of 21 classes from HMDB with spatial annotations for spatiotemporal activity recognition.

2.4 Human Activity Classification in CV

Activity classification benefits from both spatial and temporal information. Researchers have conducted different methods to capture this information. The most popular methods are recurrent neural networks (RNNs), two-stream CNNs, and 3D CNNs. Since activity classification deals with a sequence of frames, and considering the nature of RNNs that capture information through time, many of the earlier works focused on combining CNNs with Long Short-term Memory (LSTM) for the task of activity classification. For example, authors in [49] introduced the Long-term Recurrent Convolutional Network (LRCN) which feeds multiple non-overlapping frames of videos to CNNs. The output of each CNN is then fed to an LSTM network to compute class probabilities.

A more popular and successful solution to deal with the limitation of CNNs in capturing temporal information, are two-stream networks. These networks use a single RGB frame, and a sequence of optical flow (OF) frames as the input to the CNN, and then fuse the results. The OF contains information about object movements, which are essentially temporal information, and eliminates the need to use LSTM networks. Temporal Segment Network (TSN) [50] is a more

computationally efficient variation of the two-stream methods. TSN was introduced to tackle the problem of computation when processing untrimmed video clips and long-term motions. Uniformly distributed samples are taken along the temporal dimension to keep the computation cost low enough for the method to be practical. This will allow the network to capture information from the entire video without processing every frame.

3D CNNs are capable of capturing temporal as well as spatial information from stacks of RGB frames and do not require heavy computations for OF creation. The authors of [51] compared 3D versions of different residual 2D CNNs for the task of activity classification. They used a sliding window to split each video into non-overlapping video clips and input them into the 3D residual network. Activity scores of all the video clips in the video are averaged, and the activity with the maximum score is chosen for the entire video. The results indicate that ResNext-101 with 64 frames input slightly outperforms TSN.

The authors of [52] suggested using longer temporal durations to capture information from the entire video. They increased the temporal resolution while decreasing the spatial resolution to preserve the complexity of the network, as well as the required GPU memory.

2.5 Construction Workers Productivity Monitoring

One of the early attempts towards a systematic framework for productivity analysis of construction workers is [23]. The authors of [23] proposed a high-level framework to automatically interpret construction videos of cyclic micro-tasks into three different productivity information: (1) durations of activities, (2) the cyclic workflow (i.e., sequence of activities), and (3) abnormalities, such as out-of-order or excessively long activities. The activities and micro-task concepts are referred to as “task elements” and “operations” in [23], respectively. The authors of [23] first defined the sequence of activities related to the “Column Pour” micro-task. Next, they detected the concrete bucket as the most fundamental resource of the “Column Pour” micro-task using a simple classifier based on the Haar wavelets [53]. The defined activities were recognized by specifying planned locations in the image, and detecting the existence of the concrete bucket in these locations. They measured the total quantities installed, based on the number of cycles and the volume of the bucket. The same authors later expanded their work in [24] to non-cyclic construction micro-tasks. They also evaluated other simple algorithms for the detection and tracking of the objects of interest.

One of the limitations of both papers (i.e., [23], [24]) is that unless the micro-tasks are happening at completely separate locations of the site, only one micro-task can be analyzed at a time. In addition, since there is no mechanism to recognize the micro-tasks, the video should be manually trimmed into single micro-task clips and the framework should be set up for each micro-task before feeding the clips to the framework. Moreover, they used simple object detection and video understanding methods that were enough for the specific videos and the micro-tasks studied in those papers. These methods cannot be generalized to situations where the locations of activities cannot be separated or are not fixed. Newer and more powerful deep learning and machine learning methods are required for these complex situations.

There have been more recent studies on monitoring construction workers, through advanced DL-based object detection, tracking, and activity recognition methods. For example, the authors of [30] used Faster R-CNN to detect workers and construction-related objects in still site images. They constructed a relevance network based on the relations between objects and activity patterns, as well as the pixel distance of objects. Using the relevance score of detected objects, 17 different activities were inferred. Since the decision was made based on still site images, temporal information was not taken into account. They later introduced a framework in [5] to produce activity labels for all workers in 3-second video clips by considering the temporal information. They specified workers bounding boxes manually and track them separately for non-overlapping three seconds segments using a single object tracking algorithm. Next, TSN was applied to each segment to predict the probabilities for 16 classes of formwork or rebar work related activities. Furthermore, they predefined productive, semi-productive, and non-productive activities related to formwork and rebar work micro-tasks. To evaluate the productivity of the workers using the Work Sampling method, the authors of [5] calculated the ratio of the number of identified productive three seconds video segments to the total number of segments in a video. They tested this framework on a short 21-second video.

The same authors later improved different modules of their framework in [6] to identify workspaces based on the activities that are happening at different locations of the site. Aligning worker groups with workspaces in advance helps with improving the performance and safety of workers. This requires detection and understanding of dynamic workspaces. The authors of [6] first detect and track workers through 3-second video clips. A faster object detection method,

YOLOv3 (You Only Look Once) [54], together with a more efficient multiple object tracking method called Simple Online and Realtime Tracking (SORT) [55] was used for this purpose. The video clips were spatially cropped around the tracked extended bounding boxes. Next, a more accurate activity recognition network called 3D ResNext-101 [51] was fed with the resulting video clips separately to recognize 12 different actions. The action locations were then projected from the frame plane into floor plan coordinates. Assuming that action classes and their locations define workspaces, a set of rules were described to classify actions into four different workspaces. These workspaces were working areas, paths, lay-down areas, and resting areas. Finally, a density-based clustering algorithm (OPTICS [56]) was used to group area points together.

In their next research work [7], the same authors leveraged the information from nearby workers to improve the previous activity recognition framework introduced in [6] even further for the case of workers working in groups. They used YOLOv3 for workers detection, SORT [55] multiple object tracking method, and ResNext-101 to extract deep features from single worker video clips. They defined the spatial distance between workers based on the overlap and distance of detected workers' bounding boxes, and then applied the k-nearest neighbor algorithm to generate an activity graph that showed the relevance of workers to each other. Finally, they input the activity graph and the deep features, extracted from 3D ResNext-101, to a conditional random field (CRF) and inferred the most probable activity among 17 classes for each worker.

As mentioned earlier, the latest and more promising studies introduced above [6, 7], annotate their dataset and evaluate their method per-clip/segment, which is not suitable for long untrimmed surveillance videos of construction sites. The authors in [8] addressed this issue by performing "per-frame" annotation and activity recognition of excavators; however, they use simpler machine learning approaches such as Support Vector Machines (SVM) and Hidden Markov Models (HMM). In addition, all of the above papers that consider temporal information (i.e., [5–7]), either require manual worker detection, or use separately optimized detection and activity classification modules, which does not guarantee the end-to-end optimization of the method. Moreover, although these papers (i.e., [5–7]) used more sophisticated deep learning methods to improve different parts of an automatic resource monitoring system separately, their effort was limited to the recognition of activities, and they did not try to recognize the higher-level micro-tasks automatically.

Furthermore, they did not monitor the progress of the site made by workers, as it was partially done for excavators and dump trucks (e.g., [3]). Although there is a lot of research on both resource and progress monitoring separately, there are less research works on how to combine them and leverage the input-output relations to extract useful productivity data for equipment (mainly excavators and dump trucks) and even less research on how to extract meaningful and practical productivity data for construction workers. The closest research work that combined resource and progress monitoring relied only on object detection [57]. The authors computed the distance between detected workers and detected beams and columns to count the number of workers working on each product. They used the change in the size of the detected product bounding box to calculate the time of different stages (i.e., activities) of building these products. The duration and number of workers are used to calculate productivity. However, this solution does not work for all activities and all construction sites. Therefore, micro-task recognition and the integration of a resource and progress monitoring remain some of the research gaps in this line of work.

2.6 Human Spatiotemporal Activity Recognition in CV

Many spatiotemporal activity recognition methods recognize activities of each frame individually and then join them across the entire video using linking algorithms or tracking methods. Such approaches ignore the temporal information of activities resulting in poor classification and even poor localization in many situations. Action Tubelet Detector (ACT-Detector) [39] is based on the Single Shot Object Detector (SSD) [58] and uses stacks of CNN features from video frames to incorporate temporal information. It uses a set of predefined temporally replicated anchor boxes called anchor cubes, to detect action tubes in both spatial and temporal domains. The network applies two-stream CNNs to overlapping video clips to compute classification scores as well as a set of jointly regressed boxes for multiple neighboring frames. These boxes are later linked using an online linking algorithm to generate action tubes.

Depending on the duration and the nature of actions, the spatial displacement of actors in fixed anchor cubes in [39] may not be negligible. In addition, using short video clips to keep the displacement small prevents the use of longer temporal information which is needed for more confident predictions. To tackle this issue, the authors in [42] proposed Spatio-Temporal Progressive (STEP) action detector. STEP uses a progressive proposal refinement method that updates proposals (equivalent to anchor cubes) and extends their temporal duration in multiple

steps, while benefiting from the regression results of the previous steps. STEP uses a linking algorithm to connect video clip detections and generate video-level action tubes.

Most of the mentioned spatiotemporal activity recognition methods performed well for spatial detection, having acceptable frame-mAPs. However, inaccurate temporal detection decreases their overall performance, resulting in low video-mAPs. Based on evaluations in [39], 30-40% of temporal errors come from scenes where the action is going to start or has just ended. These scenes are called transitional states and are considered as negative samples when training neural networks; but due to their similarity with actions, they disturb the learned distribution of positive samples and may be classified as actions by the network during inference. Transition-Aware Context Network (TACNet) [59] tackles this issue by introducing three sets of training samples: positive, negative, and transient samples. The network not only predicts action classes (and one background class) but also decides whether or not they are transient states.

Some activities require information about the relation between different actors or the relation of actors with scene context to be recognized. Therefore, many research works have been focused on modeling these interactions (actor-actor and actor-context). Still, many activities cannot be recognized solely based on either of them. Actor-Context-Actor Relation Network (ACAR) [41] proposes a novel module to model the higher order, indirect relation between actors by leveraging actor-context first-order relations. The method shows better performance on classes that contain human interactions.

To address the task of spatiotemporal action recognition, both spatial information from the current frame and spatiotemporal information from the neighboring frames are needed. ACAR uses two separately trained networks to address this issue. However, this setup is not fully optimized. YOWO [40] proposes a network that combines these stages into a single stage and optimizes the entire network end-to-end. Moreover, YOWO uses the context from the scene around workers (not just the content inside detected bounding boxes) and gives activity labels for every frame. YOWO has two branches, a 2D CNN and a 3D CNN. The 3D CNN branch uses multiple frames as its input and the 2D CNN branch uses the most recent frame. The two branches are fused, and the last layer uses this information for both frame-level detection and activity classification.

Table 2-2 summarizes the activity classification and spatiotemporal activity recognition papers discussed in Sections 2.4 and 2.6. The activity classification methods in Table 2-2 do not have any

Table 2-2 Summary of activity classification and spatiotemporal activity recognition papers in computer vision

	Method	Context	Joint detection and classification	Temporal information mechanism	Objective
Activity classification	[49]	-	-	LSTM	Investigating the effectiveness of recurrent models for visual understanding tasks
	[50]	-	-	Two-stream	Computational efficiency
	[51]	-	-	3D CNNs	Eliminating the heavy computations of optical flow
	[52]	-	-	3D CNNs	Capturing more temporal information while preserving the complexity of the network
Spatiotemporal activity recognition	[39]	Yes	Yes	Two-stream	Using temporal information to improve activity recognition
	[42]	No	Yes	Two-stream + 3D CNNs	Progressively updating anchor cubes to track actors for better classification and localizations
	[59]	Yes	Yes	Two-stream	Reducing wrong temporal detections
	[41]	Yes	No	3D CNNs	Considering actor-actor and actor-context relations to improve activity recognition
	[40]	Yes	Yes	3D CNNs	Optimize the network end-to-end to jointly detect and classify activities

detection mechanism and require a separately optimized method such as YOLOv3 or Faster R-CNN for this step. Among the spatiotemporal activity recognition methods, ACT-Detector, TACNet, and YOWO are all jointly optimized for both detection and activity classification and use context for classification as well. However, YOWO is the only method that uses 3D CNNs instead of computationally expensive two-stream CNNs; therefore, YOWO is chosen for this research.

2.7 Summary

This chapter reviewed the literature on productivity measurement, CV-based human activity recognition, and construction workers' productivity monitoring methods. In addition, the limitations of the reviewed methods are identified and discussed.

The main findings of this review chapter are as follows:

(1) Commonly used productivity measurement methods require input data that are collected manually from the site, making them inefficient and ineffective for timely decision-making.
(2) Productivity analysis requires knowing the type of the task as defined in the daily schedule.
(3) CV-based object detection, tracking, and activity recognition can be used to automatically collect resource utilization data, such as the number of workers, their activities (productive, semi-productive, and unproductive), and the duration of their activities.
(4) The current widely used CV-based activity recognition method in the construction domain is trained and tested on trimmed video datasets, is not fully optimized, propagates error, and cannot use context from around workers to help with activity classification tasks.
(5) Activities alone are not informative enough of the type of the construction task done by workers, as some of them can be part of different tasks.
(6) There has been very little to no research on how to use workers' low-level activities to recognize higher-level tasks for productivity analysis.
(7) Although progress monitoring has been studied separately, it has been rarely combined with resource monitoring for workers' full productivity monitoring.

The next three chapters propose methods to address these limitations.

Chapter 3: CV-based Spatiotemporal Activity Recognition

3.1 Introduction

As was explained in Sections 1.2 and 2.5, the previously widely used CV-based workers' activity recognition method used by construction researchers has some practical limitations. The previous three-stage method was trained and tested on manually trimmed datasets, is not fully optimized, propagates error through its modules, and cannot use context from around workers. Therefore, this chapter proposes using a fully optimized, spatiotemporal activity recognition method such as YOWO that does not have the aforementioned limitations of the previous studies. YOWO uses a combination of 2D and 3D CNNs in a single stage, as opposed to the three-stage method. The full frames (i.e., without cropping) are used for both detection and classification of activities, allowing the method to extract useful spatial information from around workers and minimizing the effect of wrong detections on activity classification. The method outputs per-frame recognitions for every worker on both trimmed and untrimmed video clips.

Additionally, YOWO53 is proposed in this chapter to address the challenge of detecting workers who appear small in far-field construction site video frames. Moreover, a costume video dataset of six common construction workers' activities ("Standing", "Walking", "Transporting", "Drilling", "Hammering", and "Placing/fixing rebars") is manually prepared and annotated for this research. Furthermore, a semi-automatic annotation method is proposed to facilitate the time-consuming per-frame and per-worker annotation process. A detailed comparison and sensitivity analysis is conducted to prove the superiority of the method to the three-stage method.

The main tasks covered in this chapter are:

(1) Preparation and annotation of a video dataset for activity recognition of construction workers;

(2) Proposing a semi-automatic annotation method to facilitate the time-consuming per-frame and per-worker dataset annotation process;

(3) Apply YOWO to jointly detect construction workers and classify their activities into six classes of "Standing", "Walking", "Transporting", "Drilling", "Hammering", and "Placing/Fixing rebars";

(4) Proposing an improved version of YOWO (i.e., YOWO53) with better detection performance for far-field construction surveillance videos;

(5) Conducting a sensitivity analysis to compare variations of YOWO, YOWO53, and the three-stage method.

3.2 Workers Activity Recognition

The objectives of this chapter can be realized in three steps as shown in Figure 3-1: (1) dataset preparation and annotation, (2) spatiotemporal activity recognition (worker detection + activity classification), and (3) performance analysis and comparison. Each step is explained in detail in the rest of this section.

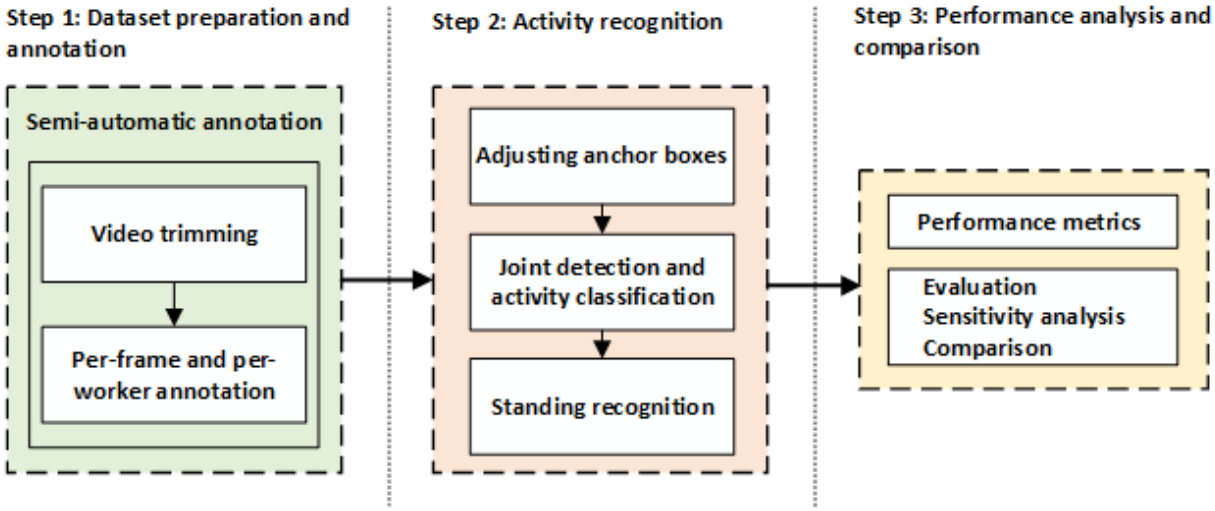


Figure 3-1 The step-by-step process of applying and comparing the joint and three-stage methods for activity recognition of construction workers

Different variations of the jointly optimized and three-stage methods are compared in this study. Table 3-1 shows how each model is referred to in the rest of this chapter based on the networks used for its 2D/3D backbones (for the joint method) or its detection/classification modules (for the three-stage method).

Table 3-1 Definition of models

Method	Models	Detection module / 2D backbone	Classification module / 3D backbone
The joint method	$YOWO_{(S)}$	Darknet-19	ShuffleNetV2_2x
	$YOWO_{(R)}$	Darknet-19	ResNext-101
	$YOWO53_{(S)}$	Darknet-53	ShuffleNetV2_2x
	$YOWO53_{(R)}$	Darknet-53	ResNext-101
The three-stage method	$Three\text{-}stage_{(S)}$	YOLOv3	ShuffleNetV2_2x
	$Three\text{-}stage_{(R)}$	YOLOv3	ResNext-101

3.2.1 Step 1: Dataset preparation and annotation

The per-frame and per-worker data annotation process is very time-consuming when done manually. The inputs of the method are short video clips containing multiple workers performing different activities continuously. The outputs are bounding box coordinates for each worker along with their activity labels for every frame of the input video clips. Even a small number of short video clips result in thousands of frames making data annotation a very time-consuming task.

To address this issue, a semi-automatic dataset annotation framework is proposed in Figure 3-2 and explained below:

(1) Video Trimming: First, video clips containing the activities of interest are trimmed from the site videos. There is no limitation on the number of workers or the number of activities that the video clip can contain (i.e., multiple workers can continuously switch between activities in each video clip). However, the trimming is done to only include the activities of interest.

(2) Per-frame and per-worker annotation: Next, frames are extracted and a small subset of them are annotated with only bounding boxes around workers. Then, an object detection network is trained to detect workers using this subset. The blue-colored boxes in Figure 3-2 correspond to this part of the framework. This network is then applied to the remaining extracted frames to obtain a full annotated dataset. The newly annotated data can later be used to improve the same object detection network and to give more accurate detections for further expansion of the dataset. Using this method, the time-consuming step of drawing bounding boxes around workers in every frame

is removed and only the labeling step is required. In addition, some of the video clips contain only one activity. These video clips can be identified during the trimming step and be labeled automatically, requiring no manual annotation. To further automate the annotation process of the remaining video clips, a simple tracking procedure can be used. Assuming that the activity of each worker does not change during the entire video clip, it is enough to label only the boxes in the first frame and use the tracking results for automatic annotation of the remaining frames. To do this, first, the location and label of boxes in the first/current frame are noted. Then, every box from the next frame is compared to every box of the first/current frame based on their IoU. It is assumed that the IoU between workers is generally lower than 0.5. Therefore, boxes with IoU higher than 0.5 are considered to belong to the same workers and are assigned the same activity labels. Finally, the current frame is replaced with the next updated frame, and the process is repeated until all the frames in the video clip are annotated. The assumption that workers do not switch between activities during the video clips or have lower than 0.5 IoU with each other does not always hold, and those frames that do not follow these assumptions should be fixed manually at the end (shown with red-colored boxes in the figure). However, this approach still helps with reducing the manual annotation effort.

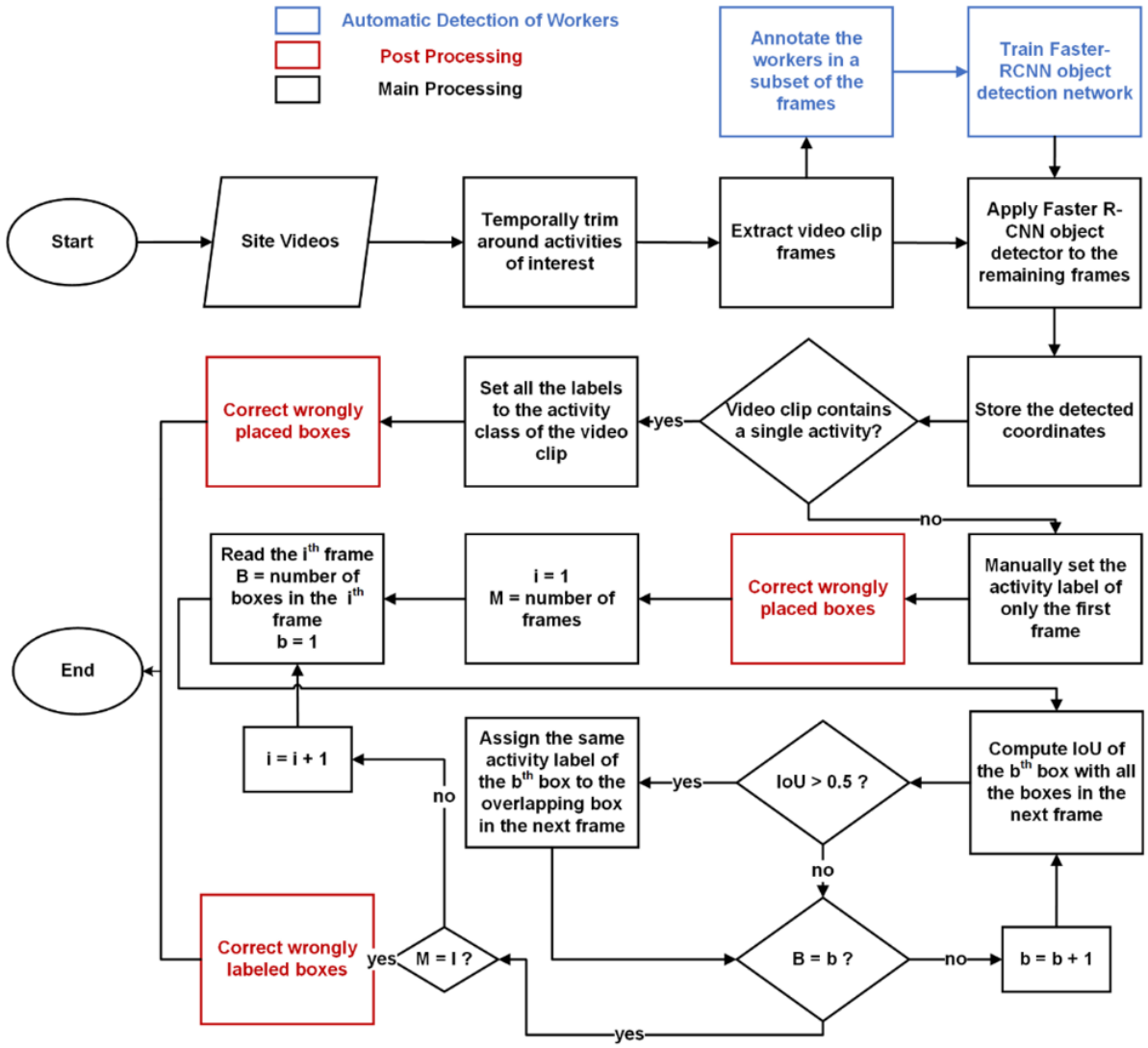


Figure 3-2 Semi-automatic dataset annotation framework

3.2.2 Step 2: Activity recognition

3.2.2.1 Adjusting anchor boxes

Most recent state-of-the-art object detectors use anchor boxes instead of directly producing the bounding box locations ([58, 54]). Anchor boxes are a set of initial bounding boxes with fixed sizes that are later modified by the object detection network to contain the objects of interest. These anchor boxes are placed at different locations of the input image, and the boxes with the highest overlap with objects of interest are chosen for training. The network gradually learns to select these boxes with a high confidence score and output a set of offsets to correct their shape and location

so that the objects of interest fall entirely inside them. The shape and size of the anchor boxes are important for both training and inference. The size of anchor boxes in the joint method depends on the size of the workers in the final feature map, which subsequently depends on the size of the input frame. Therefore, anchor boxes are adjusted accordingly.

3.2.2.2 Joint detection and activity classification

This research suggests using a slight variation of YOWO [40] called YOWO53. YOWO is a spatiotemporal activity recognition network that jointly detects and classifies the activities of every worker in every frame (per-frame and per-worker activity recognition). It requires no supervision on the content of the video clips. Inputs can be trimmed or untrimmed, contain a single or multiple worker(s), each performing a fixed activity or switching between activities. As shown in Figure 3-3, YOWO53 consists of four main blocks similar to YOWO: (1) 2D backbone, (2) 3D backbone, (3) Channel Fusion and Attention Mechanism (CFAM), and (4) Output CNN layers. Each block is briefly explained below.

2D backbone: The 2D backbone is a 2D CNN that extracts spatial information from a single frame to help with the detection of workers. The original YOWO introduced in [40] uses the YOLOv2 [60] backbone (i.e., Darknet-19) for this block. YOLOv2 is fast, but not very accurate in detecting small objects. An improved version of Darknet-19, called Darknet-53, was introduced in [54] as part of an effort to improve the performance of YOLOv3 [54] over YOLOv2. YOLOv3 showed better detection performance for small objects with a slightly lower speed. Since cameras are usually installed at a height in construction sights to cover a large field of view, workers appear very small in the site videos. Therefore, the Darknet-19 2D backbone was replaced by Darknet-53 in the new YOWO53 network to improve the detection of workers.

3D backbone: The 3D backbone is a 3D CNN that extracts spatiotemporal information from the last 16 frames (including the current frame). This branch is essential for activity classification. ResNext-101 and ShuffleNetV2_2x [61] are chosen from the different 3D CNNs that have been tested for this block in [40]. ResNext-101 is chosen because of its high accuracy, and ShuffleNetV2_2x is chosen because of its high speed. The 3D backbones for YOWO and YOWO53 models are shown in Table 3-1.

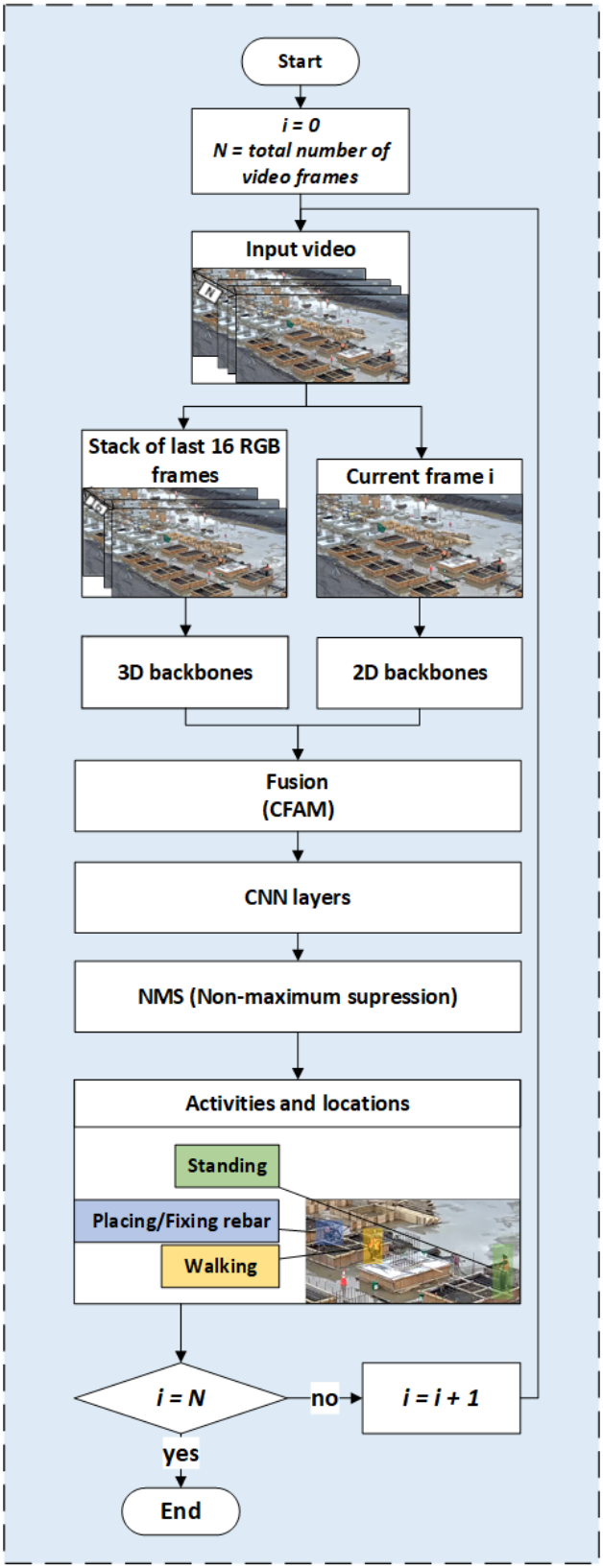


Figure 3-3 Activity recognition framework of YOWO53

CFAM: The output of 2D and 3D branches are concatenated, processed through additional convolutional layers, and fed to the CFAM block for fusion. In order to concatenate the output feature map of these two branches, they must have the same spatial resolution. In other words, the output features of these two networks must have the same receptive fields. The receptive field of a particular feature in the feature map of any layer of a CNN is the region in the input image that this feature is encoding [62]. The size of this region depends on the depth of the feature (i.e., layer) as well as the padding, stride, and kernel size of all the previous layers. Usually, as the spatial size of the feature map reduces, the receptive field of each feature increases since each feature is now responsible for encoding a larger region of the input image. The receptive fields of the final layer of both ResNext-101 and ShuffleNetV2_2x backbones used in this study and in [40] are smaller than Darknet-53. Therefore, either the spatial size of the output feature map of ResNext-101 and ShuffleNetV2_2x should be increased or the spatial size of the output feature map of Darknet-53 should be decreased. In object detection, the size of the object that can be precisely detected by the network depends on the receptive field of the output layer (detection layer) [58]. Larger feature maps have smaller receptive fields and can detect smaller objects [58]. Therefore, in this study it was preferred to increase the spatial size of ResNext-101 and ShuffleNetV2_2x output feature maps instead, to maintain the high resolution of the output feature map for better detection of small objects. To do this, a single spatial max-pooling operation was removed from the architecture of both ResNext-101 and ShuffleNetV2_2x. CFAM uses the Gram matrix-based attention to fuse the two outputs. The input of CFAM is first reshaped and multiplied with its transpose to find the correlation between different channels. Next, it goes through a SoftMax layer resulting in attention weights that are used to combine the channels. In the final feature map, each channel is summed with the weighted features of the rest of the channels. [40]

Output CNN layers: The fused feature maps are processed through additional convolutional layers followed by a regression layer. The final $H' \times W'$ output feature map has N channels to find the class probabilities, center point (x, y) offsets, height offset, width offset, and detection confidence scores for each anchor box placed at every location of the output feature map. N is given in Equation (3), and the number of anchor boxes is set to five similar to [40]. Both H' and W' in YOWO53 are 16 times smaller than the original input frame size. Additionally, Non-Maximum Suppression (NMS) is applied to remove multiple detections of the same worker.

$$N = (No.\,of\ anchors) \times (No.\,of\ classes\ +\ 4\ offset\ coordinates\ +\ 1\ confidence\ score\ ) \quad (3)$$

3.2.2.3 Standing recognition

Low performance on idling activities (e.g., standing) has been noticed in similar research works, such as [8] and [3] on excavators. This is because idling states often do not have any distinctive visual features or motions. This issue can be dealt with as a pre-processing or a post-processing step. For example, [3] used an idling recognition module to remove the idling states of excavators before applying activity classification. In this study, post-processing is preferred over pre-processing as workers perform a wide range of complex activities which may or may not involve noticeable movements. Therefore, handcrafted pre-processing methods can introduce additional errors. To this end, the average pixel displacement between every 16 frames apart standing activities of the training set is computed using Euclidean distance. This value is then multiplied by $0 < \alpha < 1$, and is used as a threshold with which the misclassified standing activities with low displacement are recognized from walking.

3.2.3 Step 3: Performance analysis and comparison

3.2.3.1 Performance metrics

Different metrics are used to separately measure the per-frame and per-worker detection, classification, and overall activity recognition (both detection and classification) performance. Additionally, activity tubes are obtained by joining detected/ground truth boxes from consecutive frames based on their activity class scores and IoU using the Viterbi algorithm [63]. The Viterbi algorithm is an efficient dynamical programming algorithm that recursively finds the most probable sequence from all possible sequences. Instead of computing the probability of all possible sequences of bounding boxes and choosing the one with maximum probability, this algorithm iteratively chooses the most probable bounding boxes in the next frame and only keeps these boxes for the next steps. The spatiotemporal activity recognition performance of the models is evaluated on these activity tubes to show how precise the model is in separating both temporal and spatial boundaries of activities.

Only detections with more than 0.5 IoU with one of the ground-truths are counted as true positive (TP) for evaluation of per-frame and per-worker classification accuracy, detection recall, overall precision, overall recall, and overall f1-score using Equations (4) to (8). Classification accuracy (Equation (4)) and detection recall (Equation (5)) measure the activity classification and detection performance of the model separately. In other words, classification accuracy ignores missed detections and is only computed for TP detections, and detection recall does not depend on the activity class of detected workers. Overall precision (Equation (6)), overall recall (Equation (7)), and overall f1-score (Equation (8)) measure the joint detection and classification performance of the model (i.e., error in both detection and classification contribute to the error of overall recall, precision, and f1-score); and are calculated for recognized activities with more than 0.25 confidence score. The confidence score of a recognized activity is the multiplication of both its detection and class confidence.

$$Classification\ accuracy = \frac{Correctly\ classified\ TPs}{Total\ TPs} \quad (4)$$

$$Detection\ recall = \frac{Total\ TPs}{Total\ groundtruths} \quad (5)$$

$$Overall\ precision = \frac{Correctly\ classified\ TPs}{Total\ detections} \quad (6)$$

$$Overall\ Recall = \frac{Correctly\ classified\ TPs}{Total\ groundtruths} \quad (7)$$

$$Overall\ f1 - \text{score} = 2 \times \frac{precision \times recall}{precision + recall} \quad (8)$$

Video-AP (video average precision) measures the area under the overall precision-recall curve of activity tubes; thus, being a good metric for evaluating the spatiotemporal activity recognition ability of the method. Video-AP is calculated for each class and different IoU thresholds. The overall precision and recall are first computed for activity tubes using Equations (6) and (7). To

find TPs, and false positives (FPs), the average IoU of all boxes in each activity tube is first calculated. This value is then multiplied with temporal intersection over union (TIoU), which is the number of valid detected boxes divided by the number of frames in the union of the ground truth activity tube and the activity tube recognized by the network. The resulting value is the 3D IoU as shown in Equation (9) [40]. If the activity class of the tube is chosen correctly and 3D IoU is higher than the predefined IoU threshold, the activity tube is considered as correctly classified TP; otherwise, it is an FP. The precision-recall curve is generated using these TPs and FPs. Next, video-AP is computed by taking the area under the precision-recall curve of activity tubes for each class and each IoU threshold. Averaging the resulting areas over all classes and further over all IoU thresholds gives the video-mAP.

$$3D\ IoU = \left(\frac{1}{No.\ boxes} \times \sum_{boxes} IoU\right) \times \frac{temporal\ intersection}{temporal\ union} \tag{9}$$

3.2.3.2 Comparison with the three-stage method

To validate that the joint method performs better than the three-stage method on untrimmed construction surveillance videos, the spatiotemporally annotated dataset of the joint method is used to prepare the training set, and validate the three-stage method proposed in [6]. The already prepared dataset for the joint method has activity labels and bounding boxes for each frame. The frames with their bounding boxes can directly be used to train the detection module in the three-stage method. However, the activity classification module of this method requires single worker trimmed video clips as its training data. This dataset is prepared automatically by tracking bounding boxes of workers in the full frames of the joint dataset for three seconds following [5], or until they switch their activities. The leftmost upper corner and the rightmost lower corner of each set of tracked bounding boxes are used to find the extended bounding boxes and to crop the full frames into single-activity, single-worker video clips. The video clips with less than 16 frames are discarded, as the network requires at least 16 frames as its input.

The trimmed dataset is used to train the classification module of the three-stage method and can be used to validate its per-clip performance as well. However, the end-to-end per-frame and per-worker detection and classification performance of the full three-stage method on untrimmed videos are what matter. The untrimmed validation set of the joint method is used for this purpose. The outputs of the detection module are again tracked for three seconds before being fed to the

classification module. Next, the activity classification module is applied on every consecutive non-overlapping 16-frame segments of the prepared video clips, and the class with maximum confidence score is assigned to all the 16 frames in each segment.

3.2.3.3 Sensitivity analysis

Surveillance cameras are usually placed at a height in construction sites to cover a large FOV. As a result, workers appear very small in the full frame. Therefore, using high-resolution frames is essential for activity recognition. However, increasing the size of the frame will reduce the speed considerably. Therefore, a sensitivity analysis is applied in this research to compare the models in Table 3-1 with different frame sizes. The comparison is done based on the introduced metrics for classification and detection performance vs. the speed (i.e., frames per second (FPS)), the size of the model (i.e., the total number of parameters), and the computational complexity (i.e., number of floating-point operations (FLOPs)) to find the best balance between these metrics.

3.3 Implementation and Results

The chosen activities and their challenges are first introduced in this section; then the implementation processes and validation results for both the joint and the three-stage methods are presented. Finally, detailed quantitative and qualitative comparisons of different models are conducted.

3.3.1 Selected activities

Table 3-2 shows the activities that are included in the dataset of this research. The videos were collected from a construction site using Axis P1425-E surveillance camera with HD resolution (1920×1080 pixels). The camera was installed on a pole at about 10 m in height. As the final goal of activity recognition in this research is productivity analysis, the method should be able to recognize and measure the durations of different value-adding and non-value-adding activities of different tasks and calculate how much time is wasted due to traveling, transportation, or idling. Therefore, three value-adding activities (hammering, drilling, and placing/fixing rebars), two non-value-adding activities (standing and walking), as well as transporting activity are chosen for this purpose, as they are regularly seen in the videos of this study. These videos are captured from the

early stage of construction of an electric substation, which was mainly focused on the construction of the foundation of this substation and included tasks such as formwork and steelwork.

Table 3-2 Selected activities and their characteristics

Trade	Activity	Mobility		Pose		Interaction with tools/objects
		Hand	Full body	Standing	Bending	
Steelwork	Placing/Fixing rebars	Yes		Yes	Yes	Yes
	Drilling	-	-	Yes	Yes	Yes
Formwork	Hammering	Yes		Yes	Yes	Yes
Generic	Standing	-	-	Yes		No
	Walking		Yes	Yes		No
	Transporting		Yes	Yes		Yes

The activities are classified based on their trades, mobility, pose, and interaction with tools/objects to identify the challenges of recognizing them using computer vision. An example of each activity is shown in Figure 3-4. Some of the activities shown in Figure 3-4 are challenging to distinguish from one another when workers are not facing the camera. For example, walking and transporting materials are very similar if the materials are not visible, since these activities have similar mobility and pose. Similarly, standing and drilling in a standing position look similar from behind when the drill is not visible. In addition, hammering is not recognizable from fixing rebars when workers are facing away from the camera. In general, activities involving some tool or interaction with an object are relatively easy to recognize as long as the tool/object is visible.

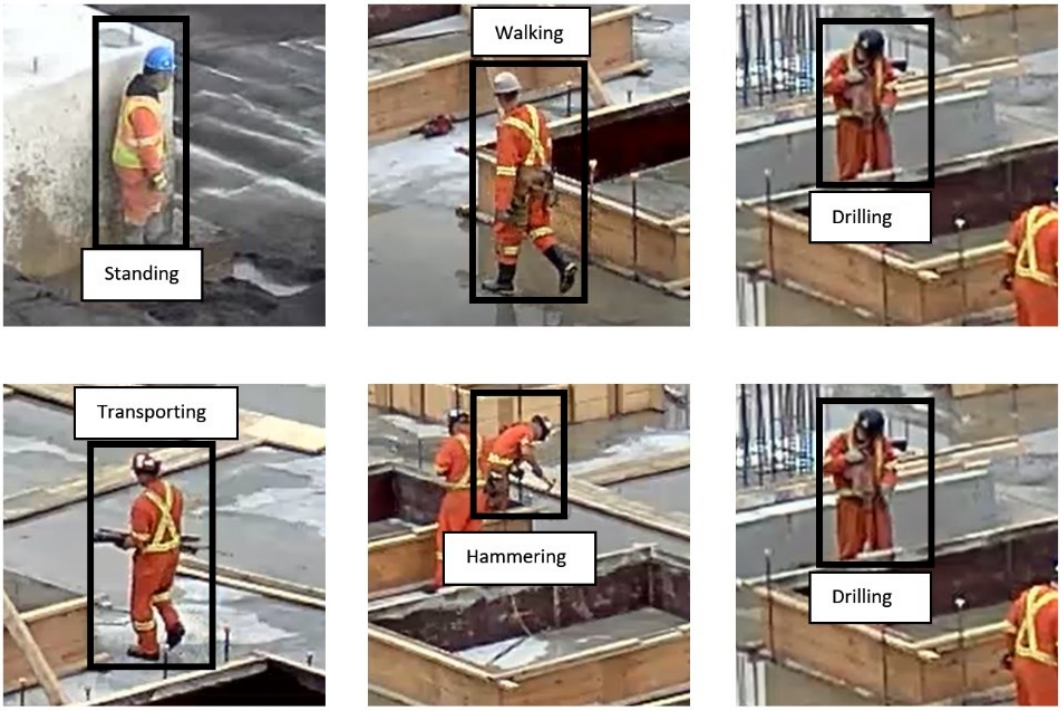


Figure 3-4 Examples of activities of workers

3.3.2 The joint method

All the models in this section are trained using PyTorch 1.8 in a UNIX environment with two 32GB Nvidia V100 GPUs. Validation is done in the same environment with a single GPU. The training batch size is set to six for $YOWO_{(S)}$ and $YOWO53_{(S)}$, while it is reduced to three for $YOWO_{(R)}$ and $YOWO53_{(R)}$ due to memory limits. In order to avoid overfitting, both 2D and 3D backbones are pre-trained on the large-scale Kinetics-600 dataset, and are frozen except for the last two layers of the 2D backbone, and the last layer of the 3D backbone. The pre-trained weights are provided by [61]. The rest of the networks are fully trained. The Darknet-19 and Darknet-53 2D backbones are pre-trained on the COCO dataset [20] and the weights are accessible on the Darknet official website [64]. All models are trained for 15 iterations and the best results achieved on the validation set are saved for comparison. Adam optimizer was used to update model weights during training.

3.3.2.1 Joint dataset

Table 3-3 shows the statistics of the dataset prepared for training and validation of YOWO and YOWO53. The original video frame size was 1920×1080. However, only the 1440×720 segments from the bottom-right corner of the frames are used to remove far-field activities as their recognition in the videos used for this study is very challenging (even to human eyes) and is outside the scope of this research. Frames are extracted from the video clips at 15 FPS (out of 30 FPS) as consecutive frames were highly similar. Since no further hyperparameter fine-tuning was done on the networks, and because of the relatively small size of the dataset, it was only split into training and validation set (i.e., no separate test set). However, all the models are compared based on their best performance on the validation set. Training and validation video clips have various durations from 2 to 10 seconds to cover activities of interest. The training dataset consists of 222 video clips (29,317 frames) and validation is done on 140 video clips (17,035 frames). Table 3-3 shows the number of video clips and frames that contain each activity. Since each video clip/frame contains multiple activities, the summation of the numbers in Table 3-3 (641 video clips and 62,508 full frames) is more than the total number of training and validation video clips/frames.

Table 3-3 Number of training video clips and full frames for each activity

Activity	Standing	Walking	Transporting	Hammering	Drilling	Placing/Fixing rebars
No. of video clips	140	155	89	93	90	74
No. of full frames	12,964	14,872	10,603	6,490	7,598	9,981

The semi-automatic annotation framework is used to annotate the video clips. First, frames were extracted and a subset of them (4000 frames) were annotated with bounding boxes around workers using LabelImg [65] annotation toolbox. The Faster-RCNN object detector was then trained using this subset and was applied to the remaining extracted frames. Faster-RCNN was chosen among state-of-the-art object detectors due to its high mAP. Next, detection results were transformed into class agnostic .xml files with the same structure used by the LabelImg toolbox. The .xml files were then opened in the LabelImg toolbox and activity labels of the first frames were updated. If

necessary, wrongly placed boxes and activity labels were manually corrected after automatic annotation of the remaining frames.

3.3.2.2 Adjusting anchor boxes

The height and width of the training workers' bounding boxes are first normalized by the height and width of the input image. Next, they are clustered with k-means clustering with IoU as the similarity measure following [40, 54]. The IoU is computed only based on the size and shape of the bounding boxes and not their locations. Therefore, there is always an overlap between boxes as they are assumed to have at least one same corner. The algorithm starts by randomly choosing five cluster centers and then assigns the rest of the bounding boxes to the cluster center with the highest IoU. Next, the cluster centers are updated by computing the average height and width of the bounding boxes in each cluster. The algorithm continues until there is no change. The clustering result is shown in Figure 3-5. The center of each cluster is used as one of the anchor boxes. These anchor boxes are then multiplied by the size of the output feature map of the network before use. YOWO uses five anchor boxes following the set up in YOLOv2, while YOLOv3 uses nine anchor boxes. However, increasing the number of anchor boxes reduces the network's speed. Therefore, five anchor boxes are used for all models, including the ones with the YOLOv3 backbone (i.e., YOWO53). In addition, since the network only detects a single object (worker) in near and mid-field, using more variations in anchor sizes is not necessary.

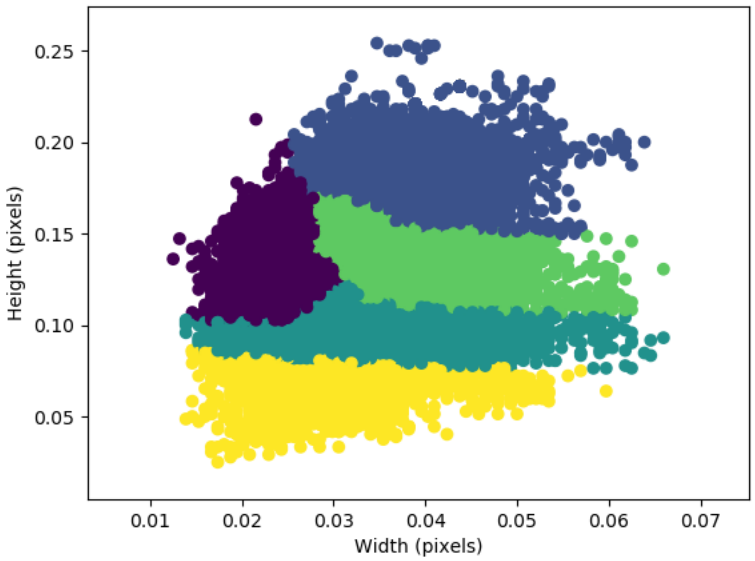


Figure 3-5 Clusters of normalized height and width of bounding boxes

3.3.2.3 Comparison and sensitivity analysis (YOWO and YOWO53)

In this section, the results using different 2D and 3D backbones and different input sizes are compared based on their speed in frames per second (FPS), size, number of floating-point operations (FLOPs), classification accuracy, detection recall, overall f1-score, and video-mAP.

Table 3-4 shows the per-frame and per-worker classification accuracy, detection recall, and overall f1-score for different models. In general, $YOWO53_{(S)}$ shows better detection recall, f1-score, and slightly better classification accuracy compared to the original $YOWO_{(S)}$ except for 896×896 input size where classification accuracy is almost the same, and 704×704 input size which shows the same f1-score for both $YOWO_{(S)}$ and $YOWO53_{(S)}$. There is at least a 2% improvement in detection recall in all cases. Some of the input sizes show even more than a 3% improvement in detection recall using $YOWO53_{(S)}$. For example, detection recall has increased from 95.6% to 98.7% for 896×896 input size when using $YOWO53_{(S)}$.

Table 3-4 Comparison of classification accuracy, detection recall, f1-score, and speed of $YOWO_{(S)}$ and $YOWO53_{(S)}$

Model	Input size	Classification accuracy (%)	Detection recall (%)	F1-score	FPS	
					Batch size	
					1	Maximum batch size
$YOWO_{(S)}$	896×896	93.0	95.6	0.886	7.9	14.4 (batch size 4)
	704×704	92.5	95.5	0.881	9.6	22.9 (batch size 7)
	512×512	91.9	92.7	0.853	11.3	48.4 (batch size 14)
	448×448	90.9	88.0	0.792	12.0	61.6 (batch size 14)
$YOWO_{(R)}$	448×448	92.0	91.4	0.847	8.9	13.6 (batch size 4)
$YOWO53_{(S)}$	896×896	92.9	98.7	0.894	5.2	5.2 (batch size 1)
	704×704	92.7	97.5	0.881	7.4	11.1 (batch size 3)
	512×512	92.6	95.8	0.872	9.3	22.2 (batch size 5)
	448×448	92.1	91.5	0.829	9.5	29.3 (batch size 7)
$YOWO53_{(R)}$	448×448	93.3	95.4	0.887	4.0	4.0 (batch size 1)

Additionally, $YOWO_{(R)}$ is tested with higher classification accuracy, detection recall, and overall f1-score (92.0%, 91.4%, 0.847, respectively) than $YOWO_{(S)}$ using 448×448 input size. Due to the

large size of the network, it was not possible to train the model with larger input sizes for a complete comparison with its YOWO53$_{(S)}$ and YOWO$_{(S)}$ counterparts. YOWO53$_{(R)}$ also shows higher classification accuracy, detection recall, and overall f1-score (93.3%, 95.4%, 0.887, respectively) compared to YOWO53$_{(S)}$ using 448×448 input size.

In all cases, fixing the 2D backbone and using a better 3D backbone (ResNext-101) resulted in both better detection, and classification performance. Similarly, using Darknet-53 instead of Darknet-19 while fixing the 3D backbones not only affects the detection performance, but also improves the classification performance. This is the exact behavior that was hoped to be seen by the integration of Darknet-53 in YOWO which resulted in YOWO53. In addition, the receptive field of YOWO53 is smaller than the receptive field of YOWO which further helps with the detection of small objects as explained earlier. These behaviors show that both 2D and 3D backbones contribute to both the classification and the detection performance of the models, with the 2D backbone requiring less GPU memory. Therefore, YOWO53$_{(S)}$ with an improved 2D backbone (Darknet-53) is preferred over using models with a better but larger 3D backbone.

Figure 3-6 and Figure 3-7 show the video-mAP calculated for YOWO53$_{(S)}$ and YOWO$_{(S)}$ respectively using 0.05, 0.1, 0.2, 0.3, 0.5, and 0.75 IoU thresholds following [40]. The video-AP for each class and each IoU threshold is shown as well. Similar to Table 3-4, increasing the input size improves the video-mAP. Comparing the result of YOWO$_{(S)}$ with those of YOWO53$_{(S)}$, YOWO53$_{(S)}$ gives a higher average video-mAP for all input sizes except for 896×896 which is a 1% lower. However, the video-mAP drops more drastically with frame size for YOWO$_{(S)}$ with a 10% drop from 0.862 to 0.768 video-mAP compared to the 5.1% drop of YOWO53$_{(S)}$ from 0.85 to 0.796 video-mAP. This shows the sensitivity of YOWO$_{(S)}$ to the frame size which is resulted from the poor performance of Darknet-19 and the larger receptive field of YOWO$_{(S)}$ which is not suitable for detecting small objects.

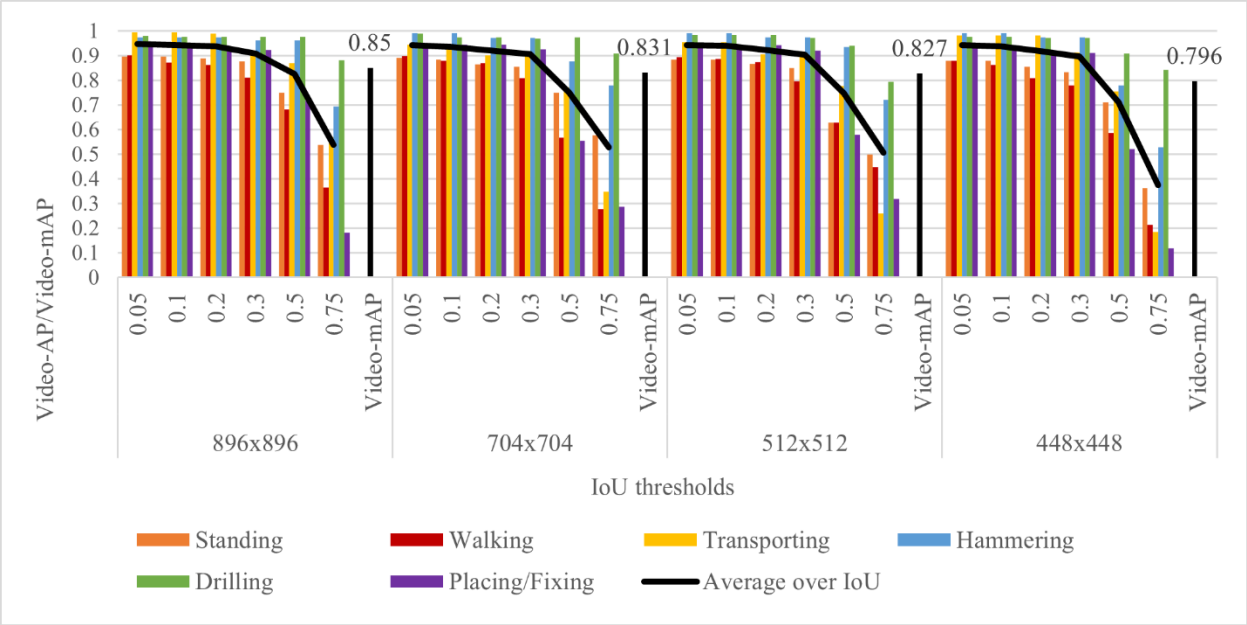


Figure 3-6 Video-mAP and video-AP at different IoU thresholds and input sizes with YOWO53$_{(S)}$

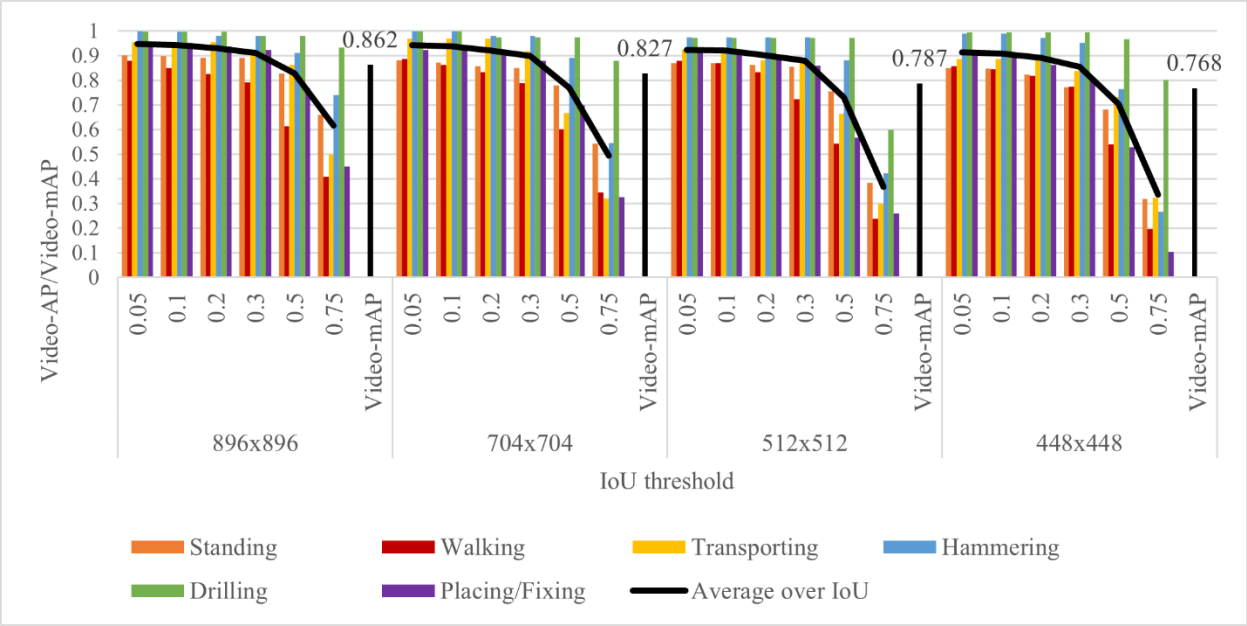


Figure 3-7 Video-mAP and video-AP at different IoU thresholds and input sizes with YOWO$_{(S)}$

The speeds of different models are shown in the last two columns of Table 3-4. Runtime speeds are calculated for both batch size one, and the maximum batch size (shown in parenthesis) that fits in the GPU. The batch size is gradually increased until it cannot be increased anymore. Smaller inputs are processed faster by the network with a batch size of one; however, they also allow the use of bigger batch sizes resulting in even faster processing. The maximum speeds achieved are

shown in the last column of Table 3-4. YOWO$_{(S)}$ is faster than YOWO53$_{(S)}$; with the largest frame size (i.e., 896×896) being processed at 7.9 FPS with a batch size of one compared to 5.2 FPS of YOWO53$_{(S)}$. However, given enough GPU memory, frames can be processed in larger batches and at higher inference speeds. For example, YOWO53$_{(S)}$ processes each 448×448 frame at 9.5 FPS, while using a larger batch size such as 7, the speed goes up to 29.3 FPS. Similarly, YOWO53$_{(R)}$ is slower than YOWO$_{(R)}$, and they are both slower than their ShuffleNetV2_2x counterparts.

Table 3-5 compares the computational complexity of different models in terms of the number of FLOPs required to process 16 frames (to be later compared with the three-stage method). The results follow the same pattern as the speeds in Table 3-4; however, the relation is not linear. YOWO$_{(S)}$ is considerably less computationally complex compared to YOWO$_{(R)}$, but has lower classification and detection performance. However, replacing the 2D backbone with Darknet-53 in YOWO53$_{(S)}$ compensates for this shortcoming with less computational overhead. YOWO53$_{(S)}$ is more computationally expensive compared to YOWO$_{(S)}$ and has about 10 million more parameters. However, YOWO53$_{(S)}$ comes with the advantage of better detection performance, which is a challenging task in construction site videos where workers appear relatively small.

Table 3-5 Comparison of different models based on the number of Giga FLOPs (16 frames) and number of parameters in Millions

Model	YOWO $_{(R)}$	YOWO $_{(S)}$	YOWO53 $_{(S)}$
Total number of parameters (Millions)	121	79	90
896×896 frames	11,195	1,812	3,145
704×704 frames	6,912	1,120	1,942
512×512 frames	3,656	592	1,027
448×448 frames	2,799	448	784

3.3.2.4 Confusion matrix

YOWO53$_{(S)}$ with the frame size of 896×896 is chosen to generate the confusion matrix as it showed the best result with 92.9%, 98.7%, and 0.894 classification accuracy, detection recall, and overall f1-score, respectively. Figure 3-8 shows the confusion matrix of this model. The matrix is generated using only the frames where all the workers were detected, resulting in 30,838 detected

workers in 12,309 frames. As was expected, the network mainly confuses the standing activity with the walking activity. This is due to the fact that workers are not entirely motionless when standing, and they sometimes take a few steps back and forth. Therefore, the standing recognition step is used in the following subsection to reduce the confusion between standing and walking activities.

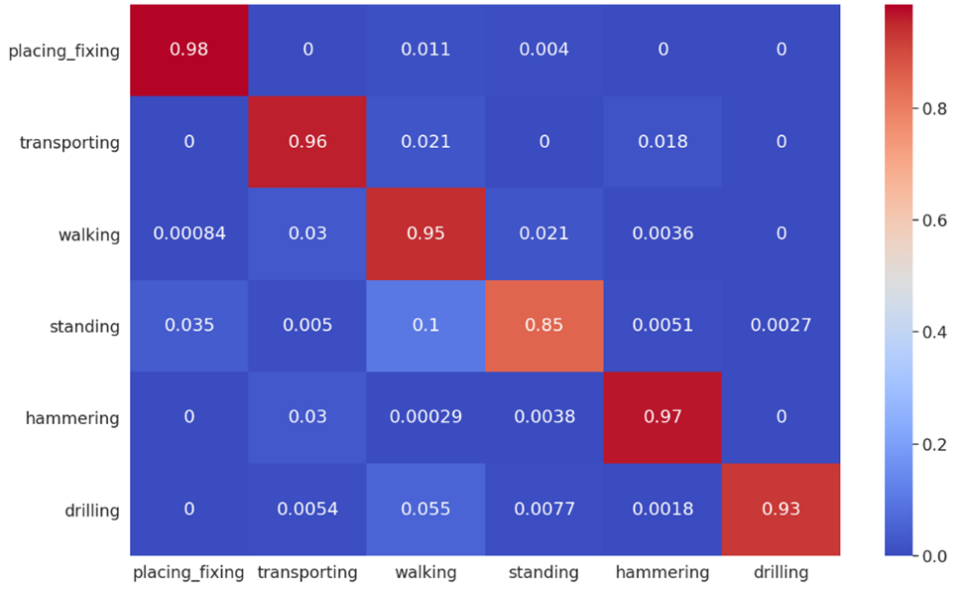


Figure 3-8 Confusion matrix of YOWO53$_{(S)}$ with 896x896 input size

3.3.2.5 Standing recognition

The average 16 frames apart displacement of standing workers in the training set is 152.4 pixels and α is chosen to be 0.03 by trial and error. Figure 3-9 shows the displacements and the average value. The confusion matrix is calculated again in Figure 3-10 after identifying the wrongly recognized walking activities. Figure 3-10 shows that the accuracy of the standing activity is improved from 85% to 91%. This approach has negatively affected the accuracy of the walking activity, reducing it by 4%; however, the overall classification accuracy is improved by 1.2% from 92.9% to 94.1%, and both walking and standing activities have acceptable accuracies above 90%.

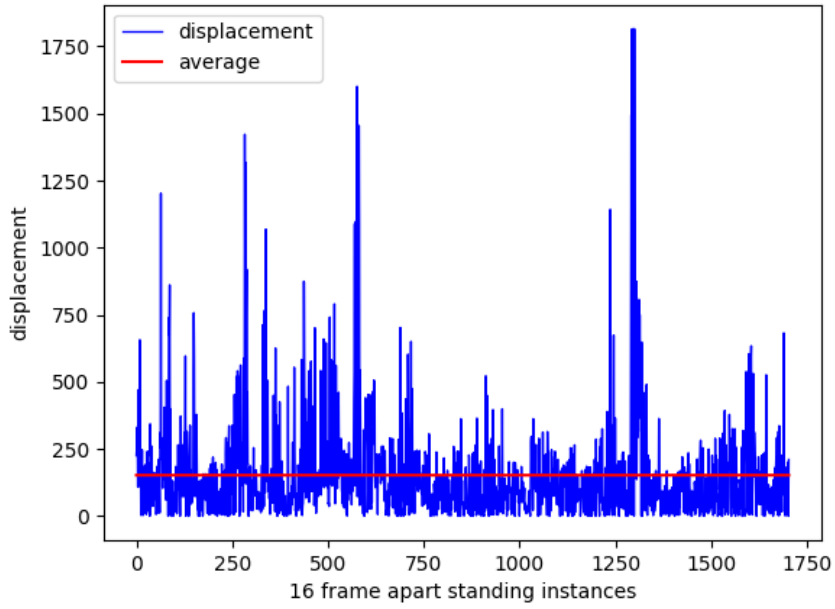


Figure 3-9 The center box displacements between every 16 frames apart standing activities

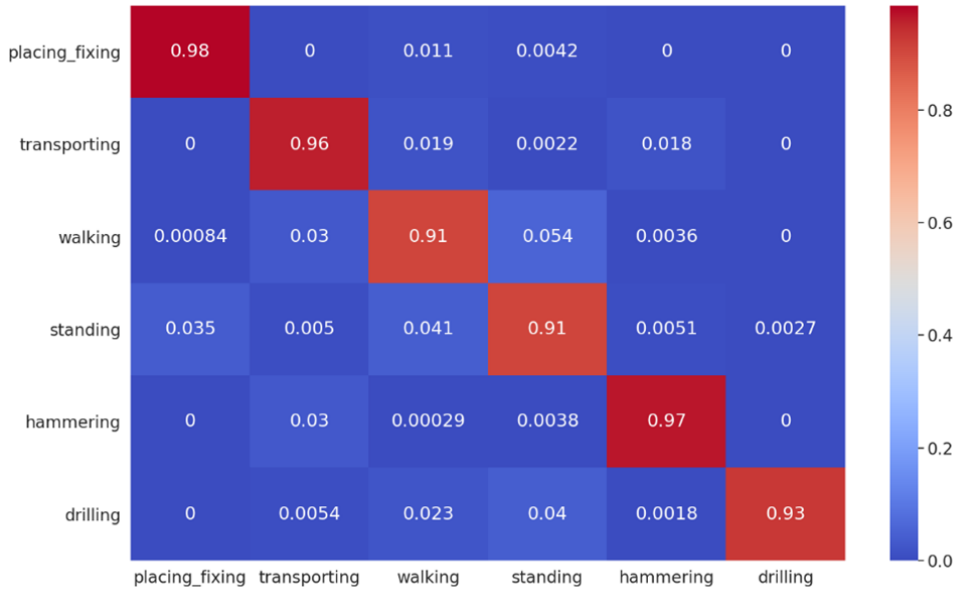


Figure 3-10 Confusion matrix of YOWO53$_{(S)}$ with 896x896 input size after post-processing

3.3.3 The three-stage method

This section first presents the performance of each module of the three-stage method separately. Next, the end-to-end per-frame and per-worker performance of the full method is presented and compared with the joint method.

3.3.3.1 Three-stage dataset

Table 3-6 shows the statistics of the activity classification module dataset. Similar to the joint dataset, frames are extracted at 15 FPS (out of 30 FPS) from the video clips. Each video clip has between 16 to 45 frames. The number of video clips in Table 3-6 is higher compared to the joint dataset as they are trimmed into shorter durations. However, the same video clips are used to prepare both datasets.

Table 3-6 Statistics of the activity classification dataset of the three-stage method

Activity	Standing	Walking	Transporting	Hammering	Drilling	Placing/Fixing rebars
No. of video clips	445	397	332	226	188	438
No. of cropped video clip frames	17,813	16,651	13,262	8,871	7,428	19,957

3.3.3.2 Detection

YOLOv3 is used as the detection module in [6, 7] because of its balance between speed and performance. Therefore, this study adopts YOLOv3 for detection as well. The same anchor boxes that were used for the joint method are used for this module as well. The network is trained for 20,000 iterations on 29,317 full frames (1440×720) and validated on 17,035 full frames with a batch size of 8. Detection precision, recall, f1-score, and frame-mAP are reported in Table 3-7 for different input sizes using 0.5 IoU threshold, and 0.25 detection confidence score.

The speed is presented in Table 3-7 for both batch size of one and the maximum batch size that fits in a single 32GB NVIDIA V100 GPU (shown in parenthesis). In some cases, higher batch sizes can fit in the GPU; however, they do not improve the speed. The number of FLOPs per 16 frame segments and the total number of parameters are given in the last two columns, respectively. The number of FLOPs is calculated for 16-frame segments since at least 16 frames are needed for the input of the activity classification module.

Table 3-7 Detection recall, detection precision, f1-score, frame-mAP, speed, and number of FLOPs of YOLOv3 with different input sizes

Input size	Detection recall (%)	Detection precision (%)	F1-score	Frame-mAP (%)	FPS		FLOPs per segment (Giga)	No. of parameters (Millions)
					Batch size			
					1	Maximum batch size		
896×896	98.6	98.3	0.984	98.67	13.3	27.2 (64)	4,847	60
704×704	98.9	98.3	0.985	99.20	20.1	44.0 (110)	2,992	
512×512	98.9	98.2	0.985	99.09	34.0	81.4 (231)	1,582	
448×448	96.9	96.5	0.966	97.01	39.9	107.6 (300)	1,211	

3.3.3.3 Tracking

The Simple Online and Real-time Tracking with a Deep Association Metric (Deep SORT) [66] multi-object tracking method is chosen as an off-the-shelf tracking module with no further tuning as it is already trained to track people. Deep SORT benefits from both the appearance and the trajectory of detected boxes to improve the tracking performance. The tracking is done frame by frame (batch size of one) at a very high speed as shown in Figure 3-11.

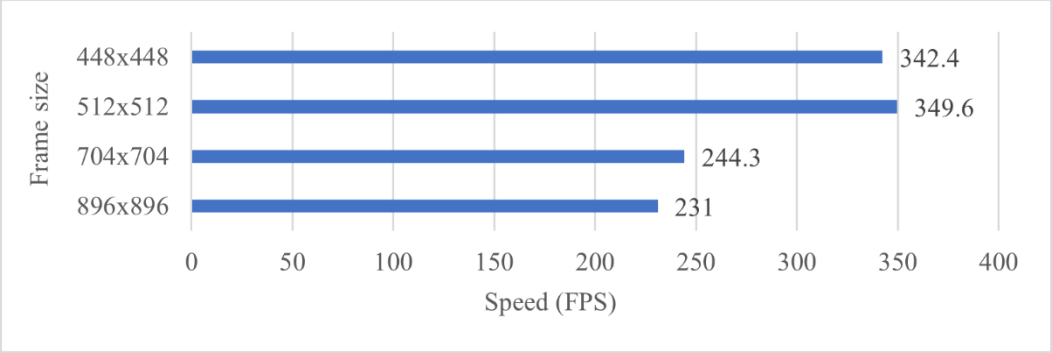


Figure 3-11 Tracking speed for different input frame sizes

3.3.3.4 Activity classification

The ShuffleNetV2_2x and ResNext-101 3D CNNs are chosen for the classification module of the three-stage method (Table 3-1) for comparison with the joint method. Both networks were pre-trained on the kinetics-600 dataset. The networks are trained using a batch size of six. The training single-worker, single-activity cropped video clips are resized to 112×112 and are then fed to the activity classification module. The speeds of the networks are presented in segment per second (SPS) instead of FPS since the networks perform classification on non-overlapping 16-frame segments. Since the input size is very small, the inference is extremely fast as shown in Table 3-8. The small size of the input frames allows using high batch sizes such as 256 with ShuffleNetV2_2x, running at 1,581.7 SPS. The number of FLOPs per segment (16 frames) and the total number of parameters for both networks are given in the last two columns of Table 3-8, respectively. As was observed before, ShuffleNetV2_2x is more computationally efficient and faster compared to ResNext-101 but with lower classification accuracy.

Table 3-8 Per-clip classification accuracy and speed of the activity classification networks

Activity classification network	Per-clip classification accuracy (%)	SPS (Segment per second)		FLOPs per segment (Giga)	No. of parameters (Millions)
		Batch size			
		1	Maximum batch size		
ShuffleNetV2_2x / (S)	70.5	4.5	1,581.7 (256)	0.358	5.422
ResNext-101 / (R)	79.9	1.4	275.8 (80)	6.929	47.533

3.3.3.5 End-to-end three-stage

The end-to-end per-frame and per-worker classification accuracy and overall f1-score of the three-stage method on the untrimmed validation set of the joint method are reported in Table 3-9. Since both activity classification and tracking modules are extremely fast compared to the detection module, the overall speed is considered to be approximately equal to the detection speed. Note that the input size of the activity classification network is fixed (112×112) and the sizes in Table 3-9 refer to the input size of the detection module. Figure 3-12 and Figure 3-13, show the video-APs of Three-stage$_{(S)}$ and Three-stage$_{(R)}$ for different IoU thresholds and different classes, as well as video-mAP for different input frame sizes.

Table 3-9 Classification accuracy, f1-score, and speed of the three-stage models

Input size of the detection module	Classification accuracy (%)		Overall f1-score		FPS	
					Batch size	
	Three-stage$_{(S)}$	Three-stage$_{(R)}$	Three-stage$_{(S)}$	Three-stage$_{(R)}$	1	Maximum batch size
896×896	68.9	76.6	0.689	0.765	3.3	27.2 (64)
704×704	65.2	74.7	0.648	0.743	0.1	44.0 (110)
512×512	61.2	71.2	0.610	0.711	4.0	81.4 (231)
448×448	59.8	68.7	0.595	0.678	9.9	107.6 (300)

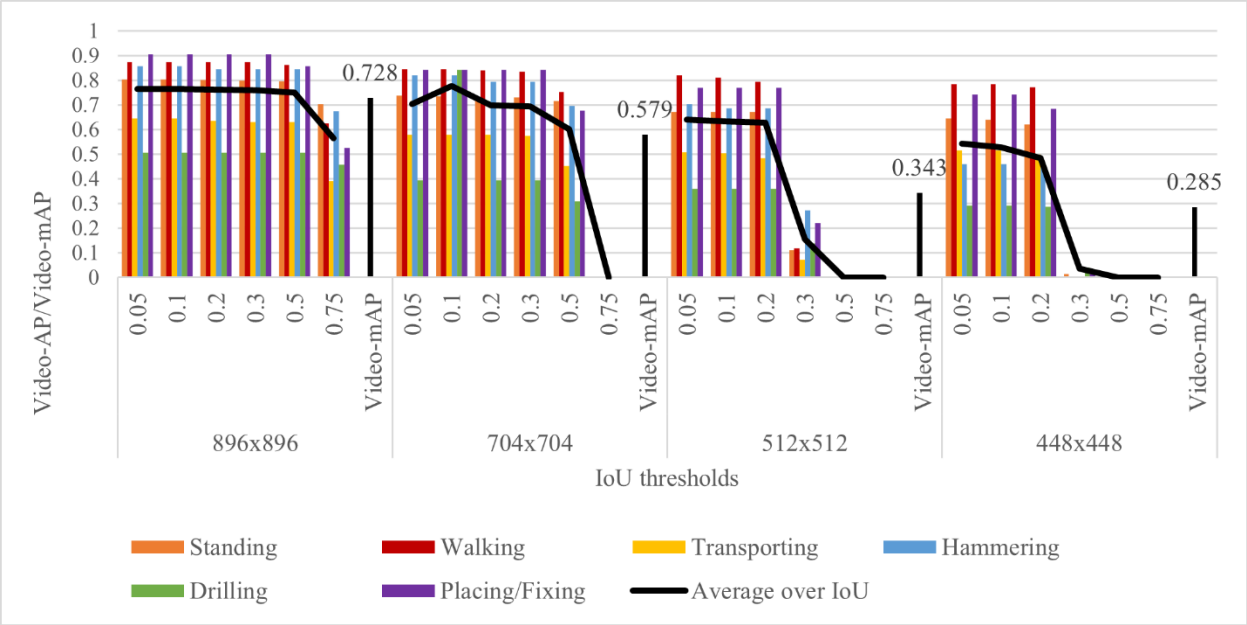


Figure 3-12 Video-mAP and video-AP of Three-stage$_{(S)}$ model for each class and IoU threshold

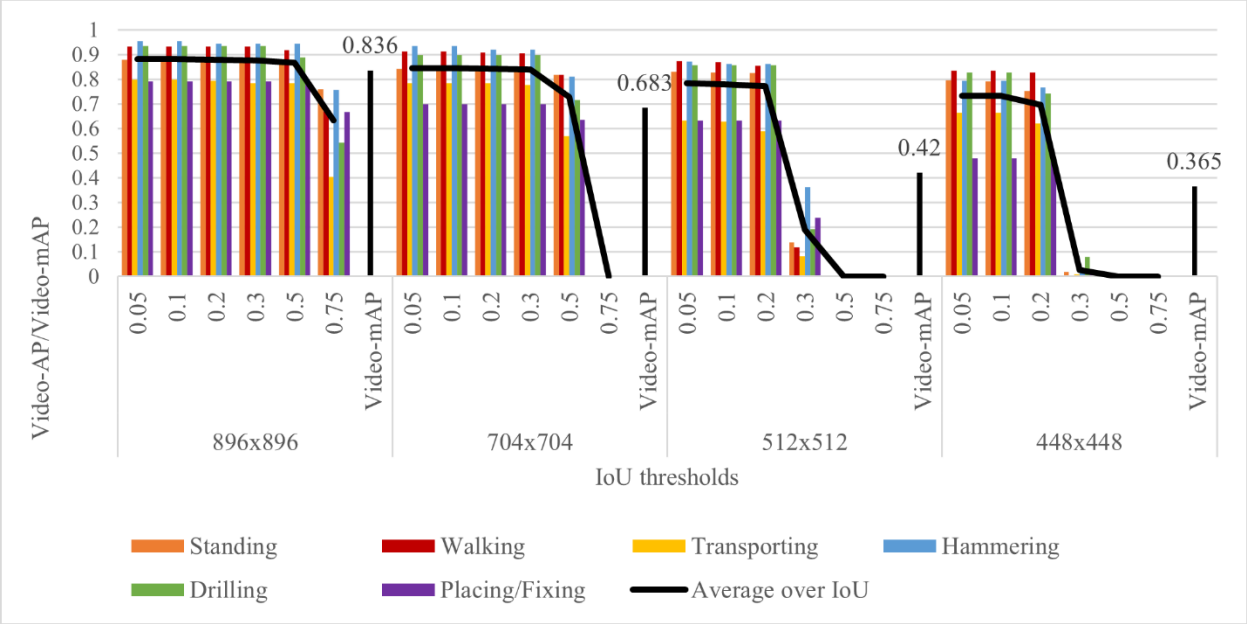


Figure 3-13 Video-mAP and video-AP of Three-stage$_{(R)}$ model for each class and IoU threshold

It is worth mentioning that the classification accuracy is only computed for detected workers; therefore, there is no direct relation between detection recalls in Table 3-7 (ratio of the number of detected workers to the number of existing workers in the video clip), and the classification accuracy. The classification accuracy depends on a combination of detection precision reported in Table 3-7, and the average IoU of detected and ground truth bounding boxes over all validation frames, which generally increases with input size. Higher average IoU between detected boxes and

ground truth boxes means better bounding box placing. Therefore, the resulting video clips, which are the inputs of the activity classification module, contain more of the workers' bodies, resulting in better classification accuracy compared to cases with either lower detection precision, lower average IoU, or both. Note that as was expected, the end-to-end per-frame and per-worker classification performance on untrimmed videos in Table 3-9 is lower than the per-clip classification performance of the classification module in Table 3-8; both because the classification module is trained on trimmed videos, and because the end-to-end performance is also affected by the errors coming from the previous (detection and tracking) modules.

3.3.3.6 Comparison of the three-stage and the joint methods

Figure 3-14 compares all the models based on their video-mAP. YOWO53$_{(S)}$ shows a higher video-mAP than all the other models in almost all frame sizes. YOWO53$_{(S)}$ has 12.5% higher video-mAP than Three-stage$_{(S)}$. The video-mAP of YOWO$_{(S)}$ is also much higher than Three-stage$_{(S)}$. Three-stage$_{(R)}$ starts very close to YOWO53$_{(S)}$ (1.4% lower) and YOWO$_{(S)}$ (2.6% lower) but drops drastically with input frame size. In general, video-mAP depends on the multiplication of two terms based on Equation (9), (1) the average IoU of all the frames in a video clip, and (2) the TIoU between detected activity tubes of a class and its ground truth tubes.

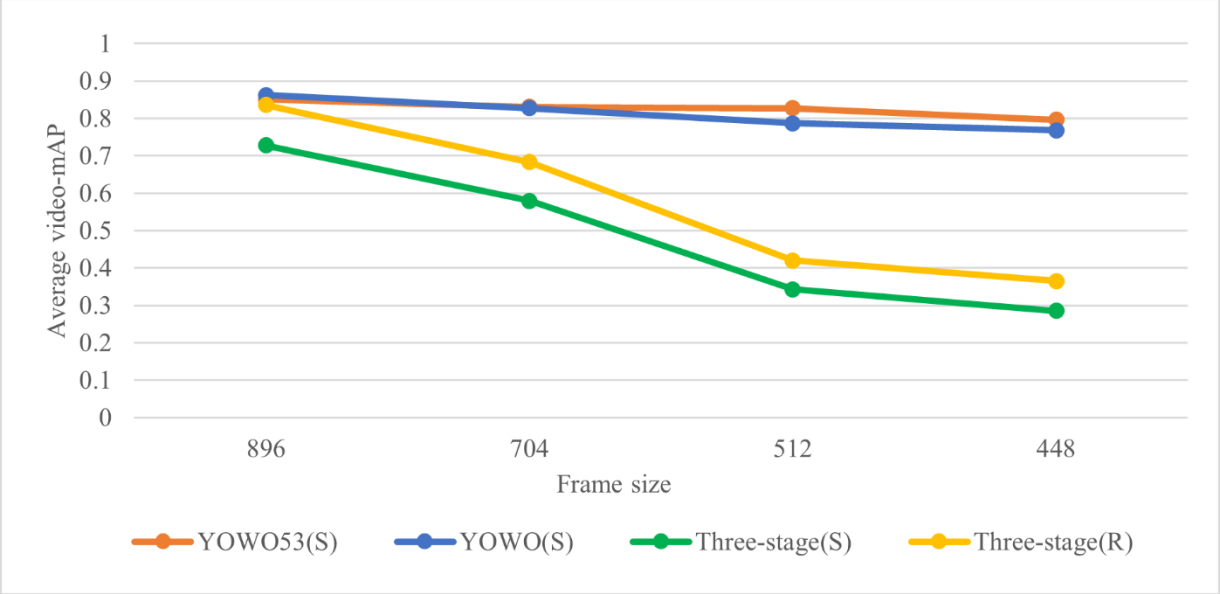


Figure 3-14 Comparison of the networks based on video-mAP

The average IoU for the three-stage method only depends on the detection module which drops with input size. The TIoU depends both on the quality of the input video clips to the activity

classification module (which is again decreased with average IoU) and the accuracy of the model on the ground truth video clips reported in Table 3-7. As a result, the video-mAP drops drastically with input size for both Three-stage$_{(S)}$ and Three-stage$_{(R)}$. On the other hand, YOWO$_{(S)}$ and YOWO53$_{(S)}$ are more robust to input frame sizes in terms of video-mAP since the activity classification does not depend on the detection results. YOWO53$_{(S)}$ is even more robust to the input size than YOWO$_{(S)}$.

Figure 3-15 compares the per-frame and per-worker classification accuracy of all the models versus their maximum speed for different input sizes. As was expected, both YOWO53$_{(S)}$ and YOWO$_{(S)}$ show much better classification accuracy (about 24% improvement) than Three-stage$_{(S)}$. YOWO53$_{(S)}$ and YOWO$_{(S)}$ are even much more accurate (about 16% improvement) than Three-stage$_{(R)}$ which uses a more accurate 3D backbone (ResNext-101). This shows the importance of joint optimization of detection and activity classification, as well as the importance of training for per-frame and per-worker activity recognition on untrimmed datasets.

As shown in Figure 3-16, the detection recall of YOWO53$_{(S)}$ with 896×896 is almost as high as the best result obtained by YOLOv3 (98.7% and 98.9% respectively). However, this number remains almost the same for YOLOv3 when decreasing the frame size (96.9% for 448×448), while it drops to 91.5% for YOWO53$_{(S)}$. In general, YOLOv3 is less sensitive to frame size compared to YOWO53$_{(S)}$, because YOWO53 uses only the backbone of YOLOv3, while YOLOv3 has additional layers to detect objects at different scales (small, medium, and large) each with their appropriate anchor boxes. YOWO$_{(S)}$ on the other hand, has a considerably lower detection recall compared to YOLOv3 (95.6% with 896×896 frames compared to 98.9%), once again showing the importance of the Darknet-53 backbone and the size of the receptive field.

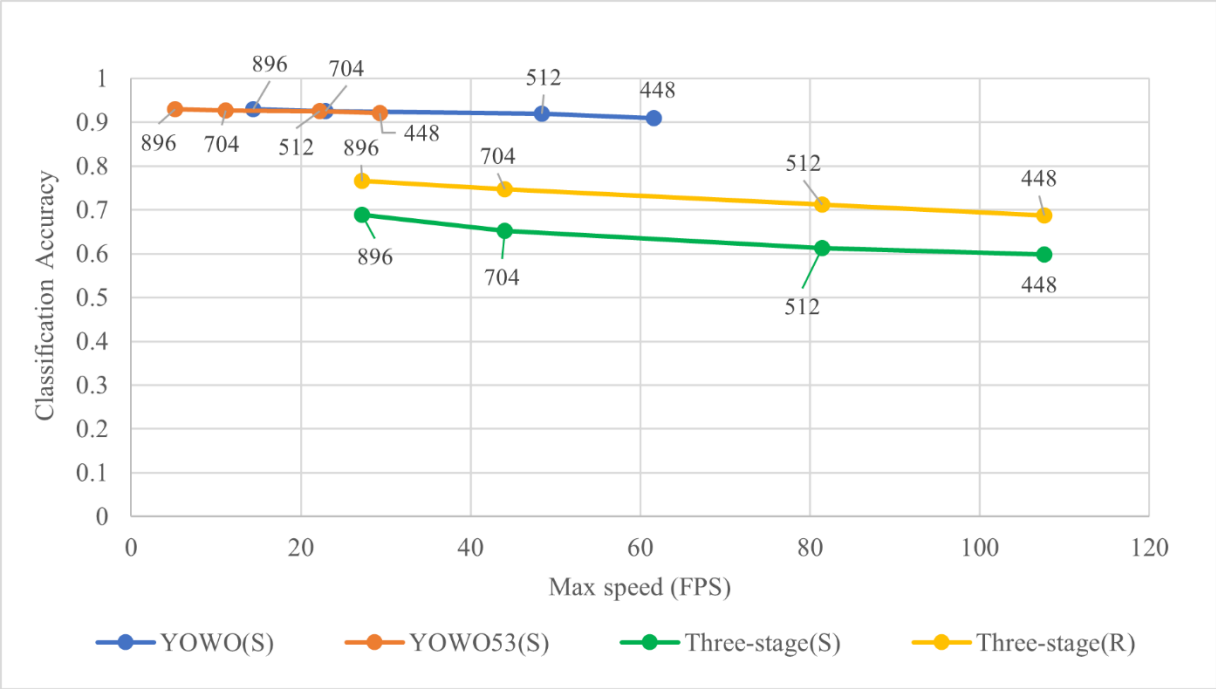


Figure 3-15 Classification accuracy vs. speed of different models

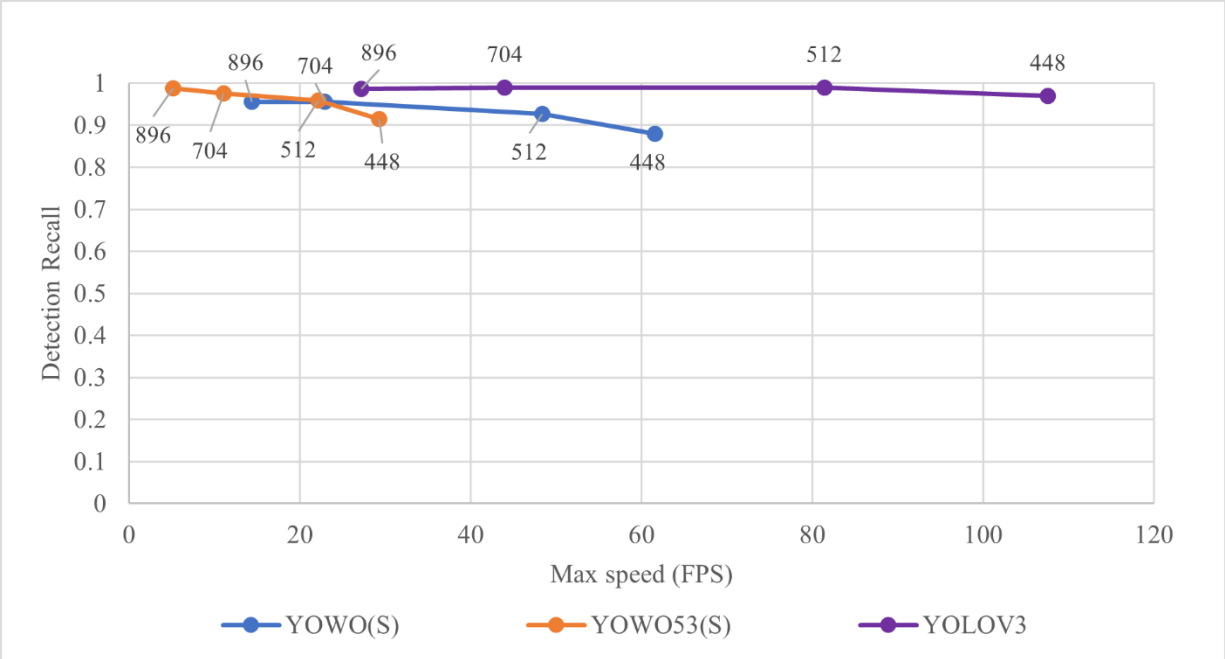


Figure 3-16 Detection recall vs. speed of different models

Figure 3-17 shows the overall f1-scores. Both YOWO53$_{(S)}$ and YOWO$_{(S)}$ show a great improvement in f1-score over the three-stage method. YOWO53$_{(S)}$ has a higher f1-score than all other models. The overall f1-score of YOWO53$_{(S)}$ is 20.5% higher than Three-stage$_{(S)}$ and 8.2% higher than Three-stage$_{(R)}$. On the other hand, the three-stage method is much faster than YOWO$_{(S)}$

and YOWO53$_{(S)}$ (27.2 FPS, compared to 14.4 FPS and 5.2 FPS respectively) as shown in the figures.

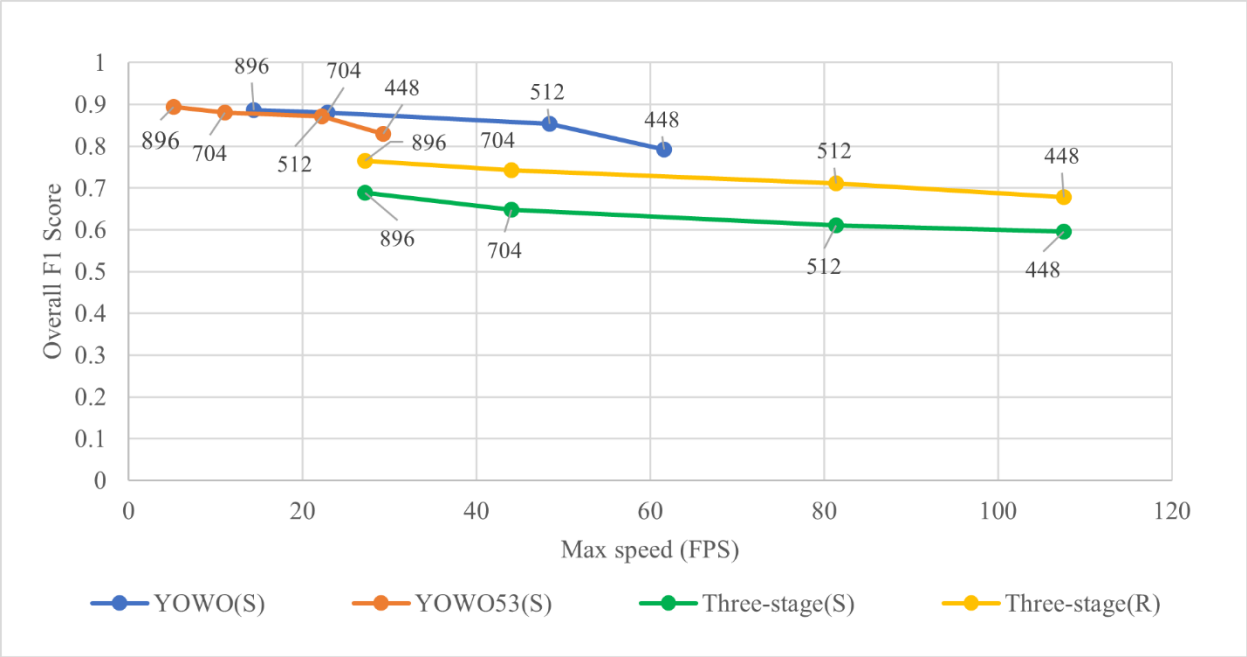


Figure 3-17 Overall f1-score vs. speed of different models

3.4 Summary and Conclusions

This chapter proposed using a CV-based method called YOWO and an improved version of it introduced in this research (YOWO53), for spatiotemporal activity recognition of construction workers. The method is jointly optimized for per-frame and per-worker detection and activity classification and can be applied on long untrimmed surveillance videos of multiple construction workers each performing different activities and switching between them continuously. The gain of joint optimization, usage of untrimmed video datasets for training and validation, and per-frame and per-worker spatiotemporal activity recognition are investigated over the previous three-stage method that required three separately optimized modules for detection, tracking, and activity classification, were trained on trimmed video datasets, and were trained and tested per-clip/segment. A sensitivity analysis considering different frame sizes and 3D backbones is conducted to compare the methods in terms of detection, classification, and overall activity recognition performance, as well as the speed, size, and computational complexity. In addition, a semi-automatic annotation framework is proposed to facilitate the preparation and per-frame and

per-worker annotation of a custom untrimmed dataset containing six common activities of construction workers.

The results show that integrating YOWO with the Darknet-53 network, and decreasing the size of its receptive field, makes the resulting network (YOWO53) better at detecting small objects with at least 2% improvement. The superiority of the joint method (both YOWO and YOWO53) over the three-stage method in activity classification is also validated with at least 16% improvement. On the other hand, the detection recall of the three-stage method (98.9%) is better than $YOWO_{(S)}$ (95.6%) and is almost similar to $YOWO53_{(S)}$ with the largest input size (98.7%). In addition, the speed of $YOWO53_{(S)}$ (5.2 FPS) is lower than $YOWO_{(S)}$ (14.4 FPS) and much lower than the three-stage method (27.2 FPS). However, for the productivity analysis purpose, 5.2 FPS is sufficient, and classification and detection performance are preferred over speed.

Chapter 4: CV-based Micro-task Recognition and Productivity Monitoring

4.1 Introduction

Although there have been many recent studies on CV-based activity recognition of construction workers, there are very few to no research works on how to use these activities for automatic productivity analysis in a practical and meaningful way. As was explained in Sections 1.2 and 2.5, the recognized activities are usually very short in duration, may be part of different construction tasks, and are not comparable to the daily schedule. Therefore, activities alone are not informative enough for productivity analysis and decision-making.

With the long-term goal of recognizing high-level construction tasks that can be compared to the daily schedule, this chapter proposes a method to combine activities into one level higher information, referred to as micro-task in this research. The proposed micro-task recognition method can detect idling as well. In addition, since productivity is measured as the relation between the input to the output of a task, resource monitoring (e.g., workers' activity and micro-task recognition) should be done along with progress monitoring. Therefore, this chapter proposes using object detection to detect completed products and compute their completion times.

Moreover, this chapter proposes a full productivity monitoring framework by combining activity recognition, micro-task recognition, and product detection. The extracted information from the aforementioned steps are combined to obtain information such as the number of workers and their locations, the duration percentages of their activities, the duration percentages of their micro-tasks, the duration percentage of idling, the number, location, and completion time of micro-task products, the average number of workers utilized for each product, as well as identification of low productivity and its underlying reasons. The main proposed modules (i.e., activity recognition, micro-task recognition, and product detection) are first evaluated on a 20-minute video. Next, the full productivity monitoring framework is applied to two case studies of 2-hour and 6-hour videos, and a detailed discussion is conducted to show how it can be used to identify low productivity, and its underlying reasons.

The main tasks covered in this chapter are:

(1) Recognizing the micro-task of construction workers from their activities;

(2) Recognizing the idling of construction workers;

(3) Detecting the completed products of micro-tasks and their completion times;

(4) Investigating the applicability of the proposed activity recognition, micro-task recognition, and product detection methods on a 20-minute video of a real construction site;

(5) Combining all proposed components from Chapter 3 and Chapter 4 to count the number of workers, calculate the duration percentages of their activities and micro-tasks, count the number of completed footing formworks, compute their completion times, find the average number of workers working on each formwork, and identify low productivity and its underlying reasons on a 2-hour and a 6-hour video of a real construction site.

4.2 Productivity Monitoring Framework

Figure 4-1 shows the overall proposed framework for resource monitoring, progress monitoring, and finally, productivity monitoring of construction workers. The proposed framework takes site videos as input and outputs different information, such as the location of workers, their activities and micro-tasks, the number, location, and completion time of built products, the average number of workers building the products, percentage of time spent on each activity and micro-task, and the percentage of time spent on each activity per micro-task.

There are three main modules (pink blocks), and one sub-modules (white block) in Figure 4-1. The main modules are: (1) activity recognition, (2) product detection, and (3) micro-task recognition. This study models micro-tasks based on a combination of activities and proposes a novel micro-task recognition method. The two remaining main modules (i.e., activity recognition and product detection) are adopted from Chapter 3 and the literature, respectively. The outputs of the main modules are either processed through the sub-module, or combined to extract different resource (input), progress (output), and productivity data. The rest of this section introduces each module and sub-module in detail.

4.2.1 Activity recognition module

YOWO53$_{(S)}$ with 896×896 input size is chosen for the activity recognition module as it showed the best performance in Chapter 3. In this section, two additional classes, one for measuring, and

the other for all other activities, are added, resulting in eight classes in total. The class "Others" is added to make sure that the model does not classify the outliers (i.e., activities that are not included in the dataset) as any of the seven main classes.

Given camera calibration matrices (i.e., intrinsic, rotation, and translation matrices), one can compute the real-world coordinates of workers with respect to an origin point $(0,0)$ from the 2D pixel coordinates (x, y) of their detected bounding boxes using Equation (10) [67], where s is the scale factor, M is the camera's intrinsic parameter, R is the rotation matrix of the camera, t is the translation vector, and finally (X, Y) is the 2D real-world coordinate. [68]

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = M(R\begin{bmatrix} X \\ Y \\ Z_{const} \end{bmatrix} + t) \tag{10}$$

The activity recognition module outputs are workers' locations, their activities, work zone heatmaps, and the duration percentage of each activity. To compute the percentages, the number of recognitions for each activity is divided by the total number of recognitions. The real-world locations can be used to obtain occupancy heatmaps for different micro-tasks or activities to understand the site's layout and locate different work zones, temporary laydown areas, and workers' traveling or transporting pathways. The recognized activities will later be used in Section 4.2.3 for micro-task recognition.

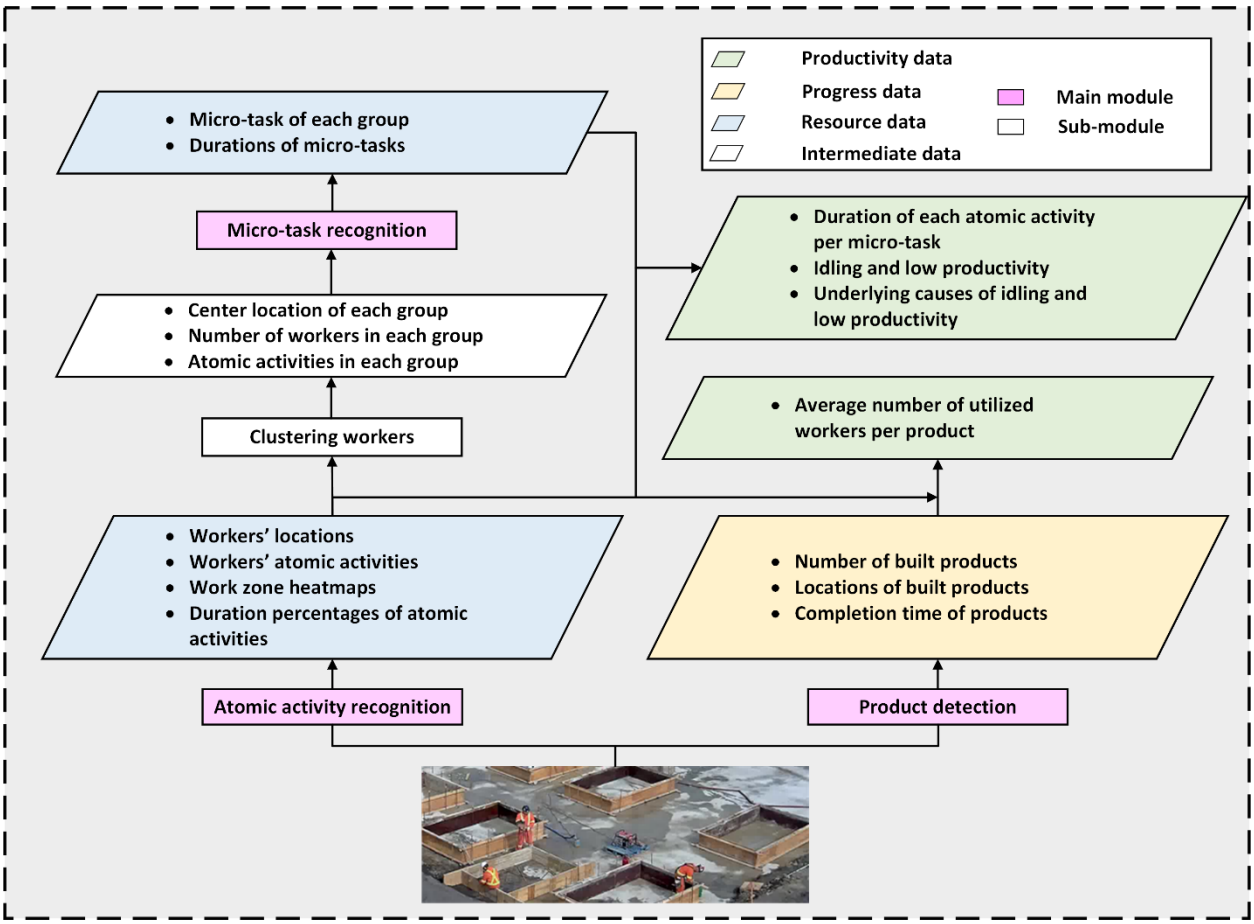


Figure 4-1 The overall resource, progress, and productivity monitoring framework

4.2.2 Clustering workers sub-module

As there are multiple workers on the site at any moment, the micro-tasks should be identified per worker. This requires tracking the workers and their activities. However, given that many workers are working in groups and are very close to each other, it is challenging to track them individually with the available methods. The fact that workers are all wearing similar clothes makes it even harder to distinguish them from one another. The authors of [24] also stated that one of their main challenges is tracking individual workers in long videos (i.e., longer than 1 minute), especially when there is occlusion, when workers have the same appearance, and they constantly move in and out of the camera's FOV. However, they also stated that productivity analysis does not necessarily require identifying the performance of individual workers. Therefore, in this study, instead of applying micro-task recognition for each worker, workers are clustered together based on the Euclidian distances between them, and the cluster micro-task is recognized using the method

explained in the next subsection. The clustering is done using a simplified greedy version of the DP-means [69] clustering algorithm.

The outputs of this sub-module are the locations of each group, the number of workers in each group, and the activities of workers in each group for every frame. The location of a group is the average location of all workers in that group.

4.2.3 Micro-task recognition module

Micro-tasks are composed of a sequence of activities. Figure 4-2 shows some of these micro-tasks and their corresponding activities. The micro-tasks are chosen based on the schedule of the site, and the sequence of activities is chosen based on the prior knowledge of the nature of the chosen micro-tasks, as well as the available videos. For example, the micro-task of "Formwork assembly" consists of walking to the staging area, transporting wooden materials, and hammering them together. However, the exact order of activities within the video frame sequence (whether the micro-task of each worker starts with walking, transporting, or hammering) and their durations are unknown. Workers may begin working on a micro-task and leave it unfinished for a while before continuing. Moreover, they may work on multiple tasks at the same time. Therefore, although there is a typical and logical sequence of activities for every micro-task, in reality, they do not follow the exact patterns.

Figure 4-3 shows the proposed micro-task recognition method. This method uses an overlapping sliding window with an initial length of $T_{initial}$ and compares the prominent activities inside the window to the order-less activities of all predefined micro-tasks (total $of\ M$ micro-tasks). The matched micro-task is chosen for the first frame (F_t) of the window. If no pattern is matched, the duration of the window T is gradually extended by T' (set as 1 minute) either until a match is found, until the number of extensions (n_{extend}) of the window exceeds some limit (N_{max}), or until the video ends. A denotes activities, I denotes the total number of activities, and D is the duration of the video in Figure 4-3. The prominent activities are obtained by counting the number of recognitions for each activity (N_{Ai}) in the sliding window, dividing it by the total number of recognitions (N_T), resulting into (P_i), and adding the ones that result in values higher than a threshold (α) into the list of prominent activities ($List_{prom}$). In addition, if more than a certain portion of the window (more than threshold β) is filled with non-value-adding activities (such as

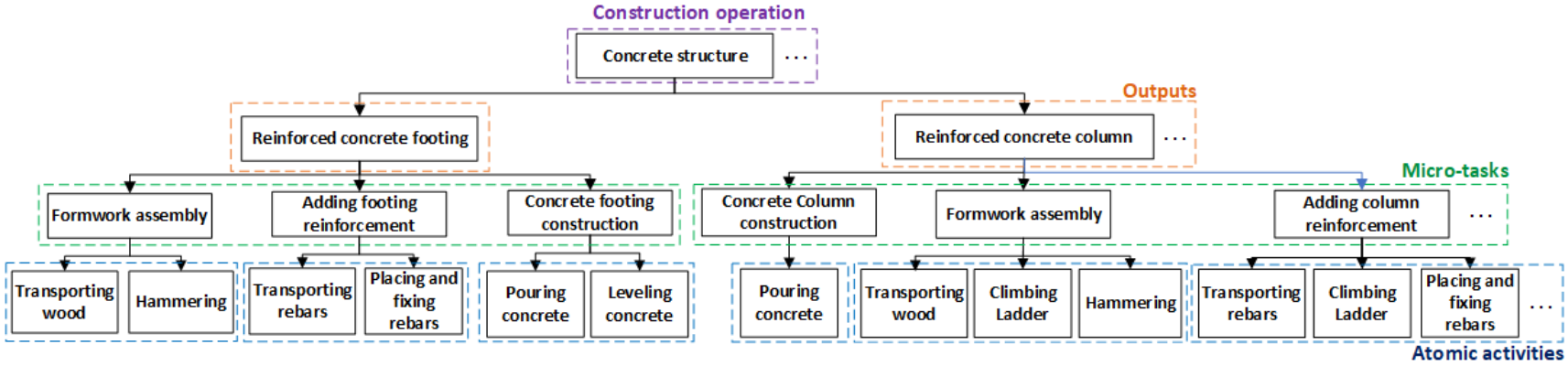


Figure 4-2 Breakdown of construction operations into outputs, micro-tasks, and activities

walking or standing), the micro-task of the first frame of the sliding window (F_t) is set to "Idling". Since workers frequently pause to take short breaks, check their work, or discuss it with their colleagues, in practice, this portion can be very high. Therefore, increasing the window length for unmatched cases results in recognizing a high percentage of "Idling" micro-tasks. This happens because the non-value-adding activities (i.e., walking and standing) appear very frequently, especially when the length of the window is increased. Therefore, although instances of value-adding activities, which are missing in the initial temporal window, are eventually found, the final micro-task is recognized as Idling, resulting in high percentages of idling for the entire input video. To fix this issue, both thresholds defined in Equation (11) for identifying the prominent value-adding activities (α), and in Equation (12) for identifying idling (β) are made to be adaptive to the number of extensions of the length of the window (n_{extend}), by using an exponential term, which becomes smaller as the length of the window increases. The Initial α and β thresholds are chosen to be 0.1 and 0.9, respectively and the second term is added by trial and error. As no ground truth was available for the micro-tasks, the percentage of unmatched micro-tasks were used as an indication of the performance of the method. The lower the percentage of unmatched micro-tasks, the better the thresholds. The limit for the number of window extensions (n_{extend}) is set to 30.

$$\alpha = 0.1 \times e^{(-0.01 * n_{extend})} - 0.001 \times n_{extend} \quad (11)$$

$$\beta = 0.9 \times e^{(-0.01 * n_{extend})} + 0.001 \times n_{extend} \quad (12)$$

The outputs of this module are the micro-tasks of each group for every frame, and the duration percentage of each micro-task. Micro-task percentages are computed similar to the duration percentages of activities. In addition to idling identification, using the percentages of activities for each micro-task, project managers can identify the underlying causes of low productivity and delays, and take the necessary actions to improve the conditions. For example, if during the assembly of formworks, the transporting activity is taking too long, this could be solved with a better site layout and placement of resources and materials. Similarly, a high walking rate can suggest a possible issue with the site layout as workers may be walking long distances to collect materials.

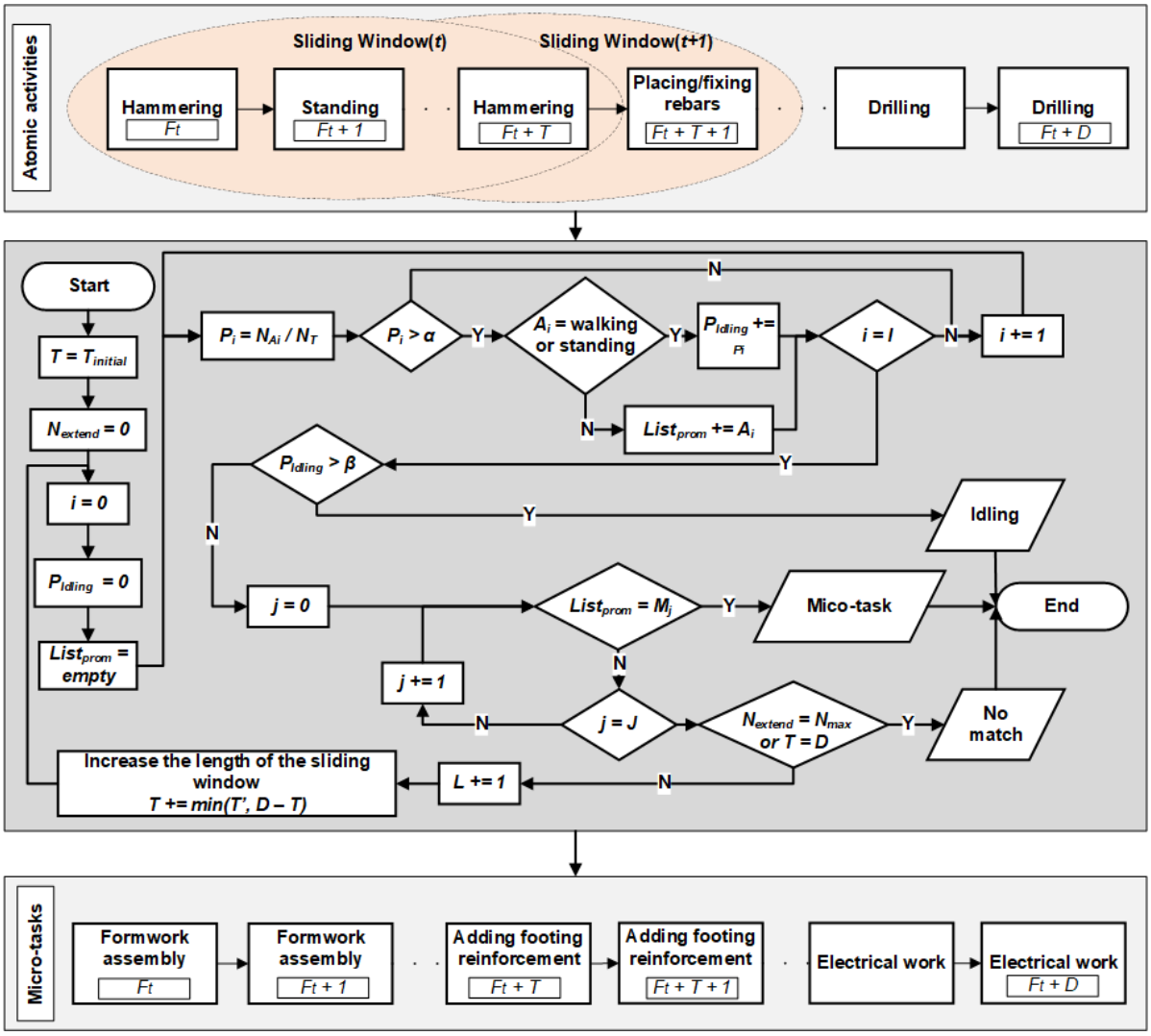


Figure 4-3 The micro-task recognition method

4.2.4 Product detection module

Different stages of construction products can be detected with different CV techniques to track the project's progress. In this study, the YOLOv3 object detection network is used for this purpose. Figure 4-4 shows the stages of building a concrete column with footing. These stages are also related to the different micro-tasks in the construction of concrete structures shown in Figure 4-2. For example, Figure 4-4(a) and Figure 4-4(b) are the initial and completed stages before and after the "Footing Formwork assembly" micro-task, Figure 4-4(c) and Figure 4-4(d) are examples of the product conditions after the "Adding reinforcement" micro-task for footings and columns, respectively, Figure 4-4(e) is related to the "Construction of concrete footing" micro-task and

(a) The initial stage of footing formwork assembly

(b) Footing formwork assembly

(c) Adding rebars of footing

(d) Adding rebars of column

(e) Footing concrete drying stage

(f) Removing formwork of footing

(g) Assembling formwork of Column

(h) Removing formwork of column

Figure 4-4 Some stages of column construction

Figure 4-4(f) is the completed stage of the "Reinforced concrete footing" product. Similarly, Figure 4-4(g) is related to the "Construction of concrete column" micro-task, and Figure 4-4(h) is the completed stage of the "Reinforced concrete column" product.

In general, multiple groups of workers are working in parallel at different locations of a construction site. To monitor productivity, the duration percentages of workers' activities and micro-tasks should be monitored along with the number of workers in each group, the result of their effort (i.e., built products), and the time it takes to achieve this result. A simplified version of this situation is when only one group of workers exists in the video, with no parallel work. Figure 4-5 shows an example of this situation where workers W_1 and W_2, working on M_1 and M_2 micro-tasks, first built product P_1 in Δt_1. They next moved to P_2, which was completed in Δt_2, and finally to P_3, which took Δt_3 to be built. As it can be seen from Figure 4-5, there is no parallel work, and workers only move to the next product after finishing the current product. In this research, as the working stage is relatively small and only one group of workers is active in many of the case study videos, it is assumed that the problem is similar to Figure 4-5. The proposed product detection module catches the end of the completion time of each of these products (i.e., t_1, t_2, and t_3) and not the beginning. These times can be computed since the frame rate and the frame number when each product is first detected are both available. Since the work is done sequentially by only one group of workers, the detection time of products is enough to calculate Δt_1 as $t_1 - t_0$, Δt_2 as $t_2 - t_1$, and Δt_3 as $t_3 - t_2$. The same assumption can be used to calculate the average number of workers it took to build each product, by dividing the total number of detected workers during the completion time of each product by their completion time.

To sum up, the product detection module outputs the number of built products, their locations, the completion time of each product, and the average number of utilized workers for each product. This information can help managers estimate the time and number of workers required to complete the project, which can be used as the input to different simulation models. Moreover, the real-world locations of detected products can be obtained as explained in Section 4.2.1.

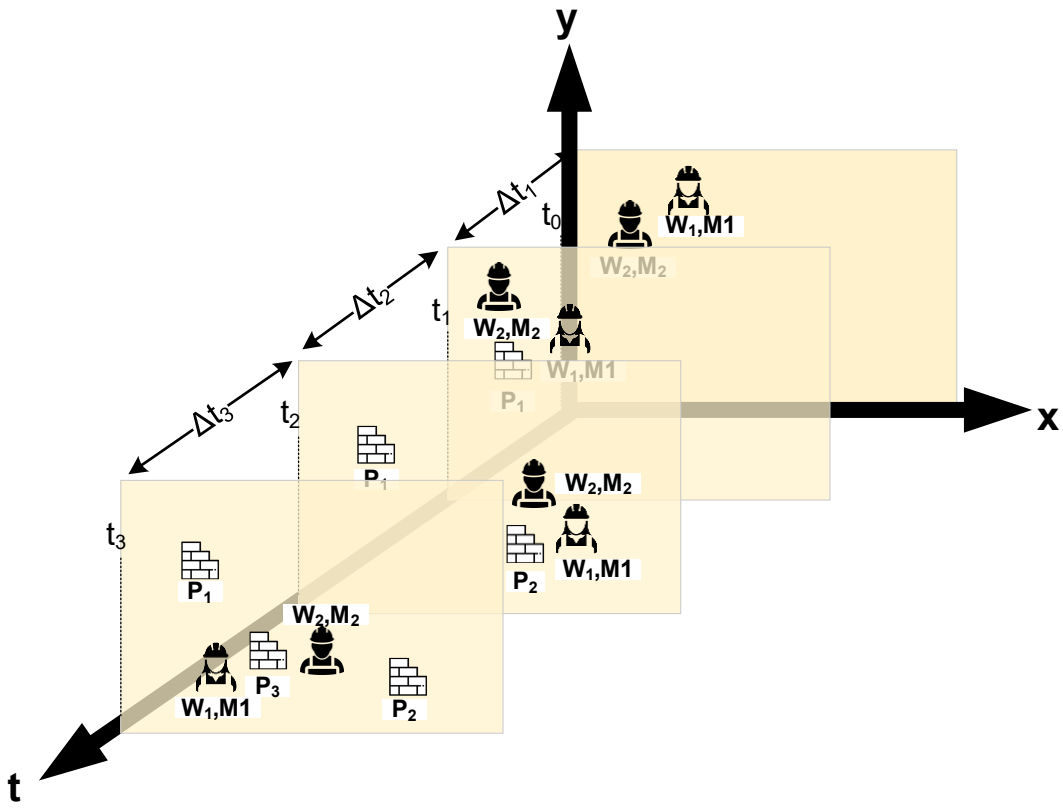


Figure 4-5 Sequential construction by a single group

4.3 Implementation and Results

The three main modules of the proposed productivity monitoring framework (i.e., activity recognition, micro-task recognition, and product detection) are first tested individually. Next, the full proposed framework is applied to a 2-hour video of "Formwork assembly" and a 6-hour video of "Formwork assembly" and "Adding footing reinforcement" to validate the applicability of the method when more than one micro-task exists in the video.

4.3.1 Activity recognition

The YOWO53$_{(S)}$ network was applied to a 20-minute video of three workers installing the reinforcement bars to recognize the activities of each worker in every frame. The video frames were extracted at 3 FPS to increase the speed, and were resized to 896×896 before being fed to YOWO53. Table 4-1 shows the duration percentage of each activity. Based on Figure 4-2, the two

activities that contribute the most to the "Adding footing reinforcement" micro-task are "Transporting" and "Placing/Fixing" reinforcement bars, which have duration percentages of 10.7% and 33.9%, respectively. As was expected, Table 4-1 shows that workers spend a lot of time walking or standing between these two value-adding activities. Based on manual observation of the video, these instances of walking and standing are not idling, and are related to the nature of human labor as workers need to take regular short breaks and pauses. The two unrelated "Hammering" and "Drilling", based on Figure 4-2, activities are not recognized in the video, which shows the good performance of $YOWO53_{(S)}$. The existence of the "Measuring" activity was also manually verified from the video, however, it is not a part of the "Adding footing reinforcement" micro-task, and was occasionally done by another worker on some of the already built footing formworks for quality control.

4.3.2 Micro-task recognition

The method introduced in Section 4.2.3 is used to recognize the micro-tasks from the recognized activities in Table 4-1. The initial length of the sliding window is set to 5 minutes with 3 FPS and N_{max} is set to 30. If there is no match after 30 times of window extension, the micro-task is recognized as "Unmatched" and is included in the "Not defined" micro-task. The "Not defined" micro-task is also chosen when the prominent activities are "Measuring", "Drilling", or "Others", since they are not defined as a part of any specific micro-task in this case study. However, based on the type of the project, they can be combined with other activities to define a specific micro-task similar to "Hammering", "Placing/Fixing", and "Transporting". The thresholds α and β are initially set to 0.1 and 0.9, respectively, and are exponentially reduced as the length of the window increases.

Table 4-2 shows that the proposed method for micro-task recognition correctly recognized the main micro-task from the two potential micro-tasks of "Adding footing reinforcement" and "Formwork assembly", with 71.3%. As was manually confirmed before, most of the non-value-adding "Walking" and "Standing" activities were not in fact idling (2.3%), and were due to the nature of human activities.

4.3.3 Product detection

For the evaluation purpose, only footing formworks are considered as the products of interest in this study. Table 4-3 shows the statistics of the dataset prepared for the detection of footing formworks. The initial size of the images was 1440×720, and they were resized to the same size as the input of YOWO53$_{(S)}$ trained models (i.e., 896×896). Table 4-4 shows the detection recall, precision, f1-score, and mAP at 0.5 IoU of YOLOv3, which was trained for 20000 epochs with a batch size of 6. Although the performance is already good, a series of post-processing operations were applied to reduce the chance of noisy false positives and false negatives in the case study in Section 4.3.4.3. For example, since the formworks are built permanently, once a completed formwork is detected, it must remain in the same location for the rest of the video. This step is applied to detect the formworks that were detected in the previous frames but are now missing due to occlusion by workers or YOLOv3 errors. In addition, once a new formwork is detected, it is assumed that it must be detected in 90% of the next 300 frames as well; otherwise, it is considered a false detection. Using these additional post-processes, noisy detections can be eliminated when applying YOLOv3 to long surveillance videos of construction sites.

In addition, the real-world location of detected products and workers are obtained using their pixel locations and camera calibration matrices. The QtCalib software [70] is used in this study to obtain the calibration matrices from a single camera. The software uses the Tsai algorithm [71] to compute the calibration matrices from the pixel and real-world dimensions of a rectangular object such as footing formworks. The real-world dimensions of formworks were obtained from the project plan, and the pixel dimensions were obtained from video frames. It is assumed that all the objects (i.e., workers or products) are at the same height ($Z_{const.}$), which can be set to 0 in Equation (10) for simplicity. There are many different calibration methods based on two cameras or chessboard patterns that can be used in the future for better localization accuracy. However, this simple method is chosen for this study since localization is not the main objective.

Table 4-1 Activity duration percentages in a 20-minute video of adding footing reinforcement

Activity	Duration percentage (%)
Standing	26.6
Walking	15.4
Transporting	10.7
Measuring	11.5
Hammering	0.0
Drilling	0.0
Placing/Fixing rebars	33.9
Others	1.9

Table 4-2 Micro-task duration percentages in a 20-minute video of adding footing reinforcement

Micro-task	Duration percentage (%)
Formwork assembly	0.0
Adding footing reinforcement	71.3
Idling	2.3
Not defined	26.4

Table 4-3 Footing formwork detection dataset

Input size	No. of training frames	No. of validation frames	Total training formwork instances
896×896	5,479	509	33,802

Table 4-4 Detection recall, precision, f1-score, and mAP of YOLOv3 on footing formwork dataset

Input size	Recall (%)	Precision (%)	F1-score	mAP@0.5 (%)
896×896	98.0	98.0	0.98	98.54

4.3.4 Case 1 – single micro-task

The full productivity monitoring framework in Figure 4-1 was applied to a two-hour video of mainly footing formwork assembly.

4.3.4.1 Activity and micro-task recognition

The duration percentages of activities in the full video are shown in Table 4-5. Based on these results, workers spent 67% of their time "Walking" and "Standing", which are non-value-adding activities. However, this does not necessarily mean that workers were mostly idling, because not all of these "Walking" or "Standing" instances are occurring continuously. As was explained before, to distinguish idling from short pauses between value-adding activities, only long instances of continuous walking and standing are considered idling. In addition, the two main activities of the "Formwork assembly" micro-task (i.e., "Hammering" and "Transporting") were recognized with 9.6% and 13.8% duration percentages, while the remaining unrelated activities (i.e., "Measuring", "Placing/fixing", and "Others") were recognized with very low percentages. Moreover, 8.4% of workers' time was spent on "Drilling". This activity was not included in Figure 4-2 as part of any pre-defined micro-task; however, its existence was later visually verified from the video.

The duration percentages of activities alone do not give useful information for productivity analysis. The first row of Table 4-6 shows the micro-task duration percentages for the full video, and the remaining rows show the activity duration percentages for the micro-task written at the top of the columns. Based on the first row of Table 4-6, the main micro-task was "Formwork assembly" (80.8%) during these two hours, while idling was 17.3% of the video. Although workers spent most of their time on a value-adding micro-task, they may not have been fully productive during this micro-task. The first column of Table 4-6 shows that 61% of the "Formwork assembly" micro-task consisted of the "Walking" and "Standing" activities, 11.2% of this micro-task consisted of "Transporting" activity, and only 17.7% were recognized at "Hammering". Note that other activities are also recognized as part of the "Formwork assembly" micro-task; however, they were not recognized as the prominent activities by the micro-task recognition method, as they make up relatively smaller portions of this micro-task. For the "Formwork assembly" micro-task, it is expected to see a lot of "Standing" in-between value-adding "Transporting", and

“Hammering” activities, as workers usually pause and take short breaks after regularly bending down to hammer or pick up materials. Because of the nature of this micro-task, it may not be possible to reduce the “Standing” activity much. On the other hand, although some of the recognized “Walking” instances are because of the nature of the micro-task; the rest may be related to low productivity. Therefore, it is important to recognize the micro-task before deciding on the underlying causes of low productivity. Additionally, although “Transporting” is expected to be seen during the “Formwork assembly” micro-task, it is preferred to be relatively less frequent than the value-adding “Hammering” activity. Reducing the “Transporting” activity, naturally reduces the “Walking” activity as well. As explained in Section 4.2.3, although relatively high percentages of “Walking” and “Transporting” are not considered as idling, it can indicate possible issues with the site layout or the location of the laydown areas. Both idling and high “Walking” and “Transporting” percentages are later investigated in Section 4.3.4.4.

Table 4-5 Activity duration percentages in a two-hour video on the 25th of September 2019 (case 1)

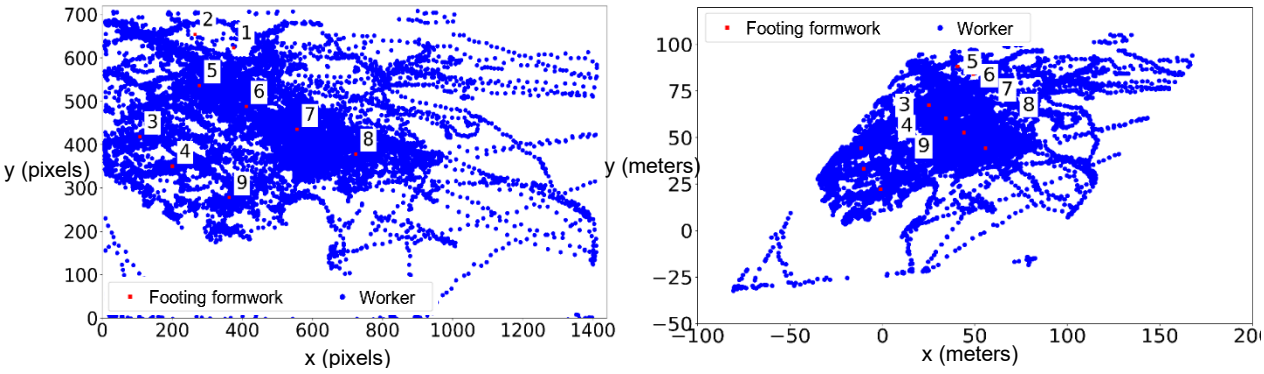
Activity	Duration percentage (%)
Standing	25.7
Walking	41.3
Transporting	9.6
Measuring	0.7
Hammering	13.8
Drilling	8.4
Placing/Fixing rebars	0.3
Others	0.2

Table 4-6 Micro-task duration percentages in a two-hour video on the 25th of September 2019 (case 1)

Micro-task / Duration percentages	Formwork assembly (%)	Adding footing reinforcement (%)	Idling (%)	Not defined (%)	Gross total (%)
Total	80.8	0.0	17.3	1.9	100
Standing	20.4	0.0	45.1	42.2	
Walking	40.6	0.0	44.8	45.5	
Transporting	11.2	0.0	3.9	11.3	
Measuring	0.5	0.0	0.2	0.1	
Hammering	17.7	0.0	0.7	0.9	
Drilling	9.6	0.0	4.8	0.0	
Placing/Fixing rebars	0.3	0.0	0.2	0.0	
Others	0.1	0.0	0.3	0.0	
Gross total (%)	100	NA	100	100	

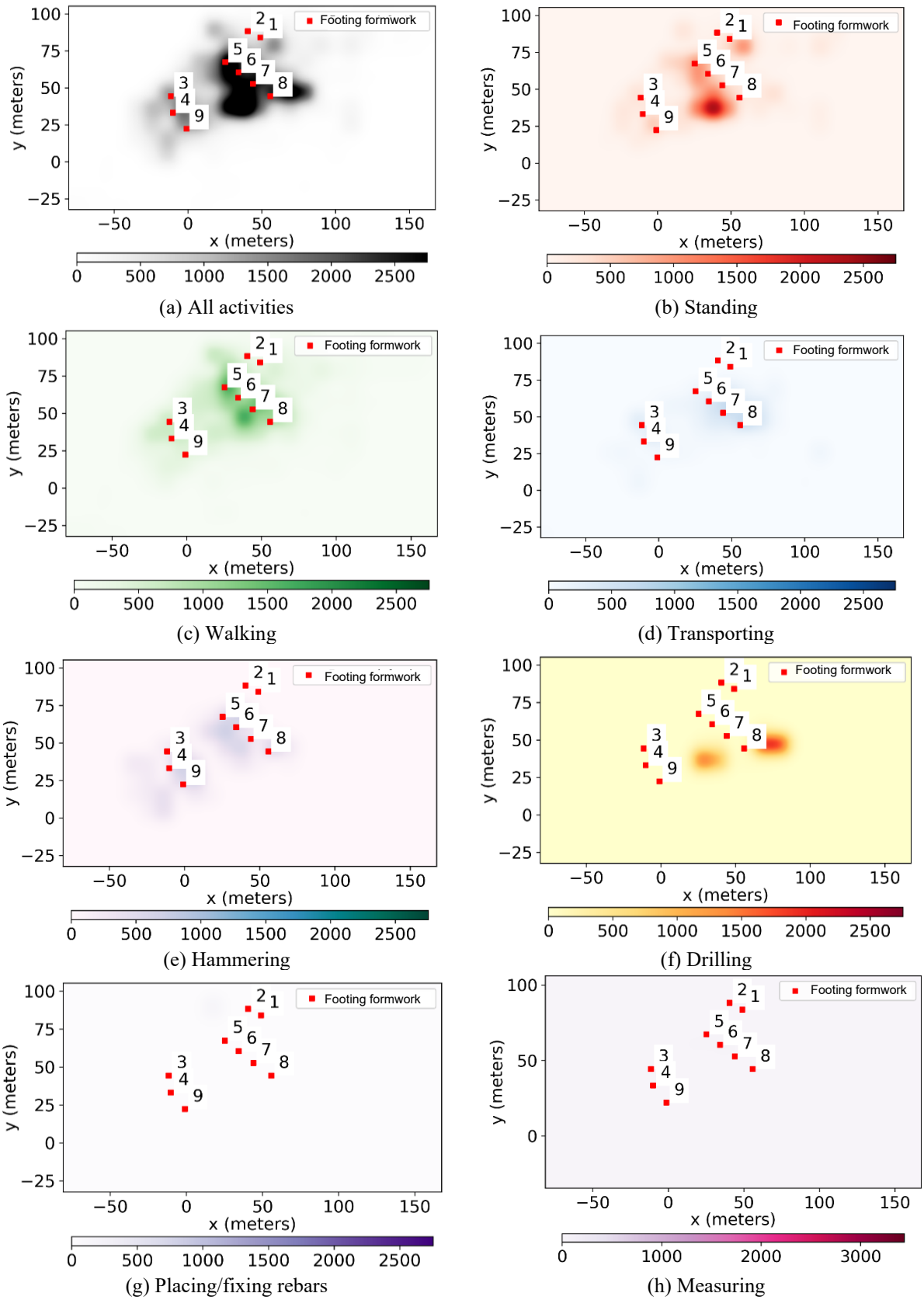
4.3.4.2 Generating heatmaps of activities and workspace localization

Location of activities can be used to obtain a heatmap for each activity to better understand the work zones, laydown areas, and transportation or traveling paths. The blue points in Figure 4-6(a) show the pixel location of workers in the full two-hour video. The red squares show the center pixel location of detected footing formworks during this period. The IDs of the formworks show the order in which they were detected by the product detection module. Figure 4-6(b) shows the real-world locations of workers and detected footing formworks, which are obtained using Equation(10). Using the real-world locations, a separate heatmap is generated in Figure 4-7 for each activity. The heatmaps show the same information in Table 4-5 in a qualitative and easy to visualize format. In addition, these heatmaps can help identify different work zones. For example, Figure 4-7(e) shows that the "Hammering" activity mostly took place where the footing formworks were planned to be built, while the "Drilling" activity was limited to two small areas shown in Figure 4-7(f). The heatmaps can also be used to localize laydown areas and identify possible issues with the site layout, which will be explained in detail in Section 4.3.4.4.



(a) Workers' pixel coordinates (b) Workers' real-world coordinates

Figure 4-6 Workers' locations during the two-hour video on the 25th of September 2019 (case 1)



(a) All activities

(b) Standing

(c) Walking

(d) Transporting

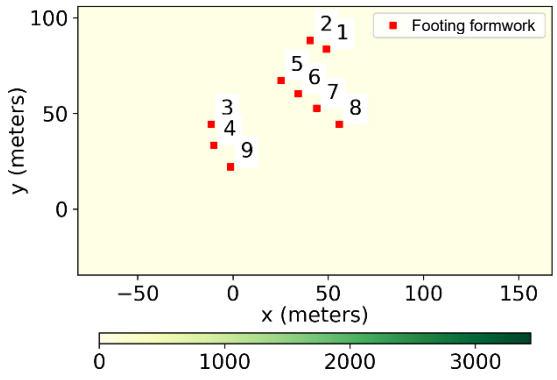
(e) Hammering

(f) Drilling

(g) Placing/fixing rebars

(h) Measuring

Figure 4-7 The heatmap for each activity (case 1)

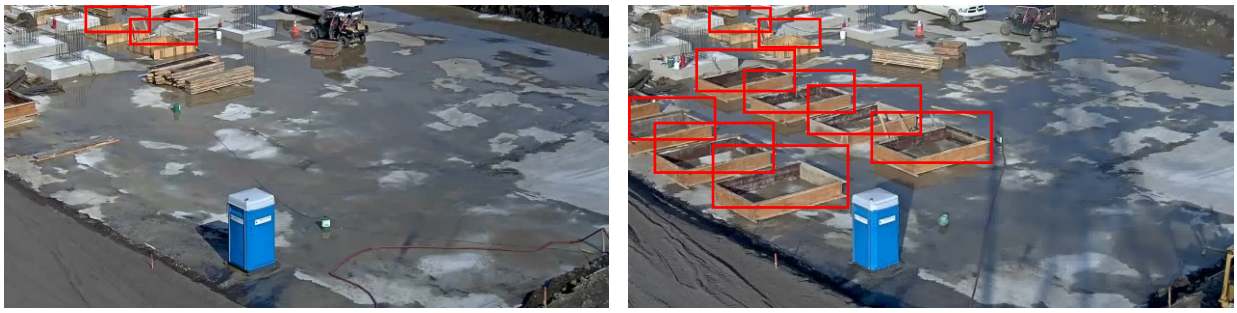


(i) Others

Figure 4-7 The heatmap for each activity (case 1) [continued]

4.3.4.3 Progress monitoring

In addition to monitoring workers, the work progress can be monitored by detecting completed products such as footing formworks. There were two already completed footing formworks at the beginning of the video, shown in Figure 4-8(a), and a total of seven footing formworks were completed by the end of the two hours, as shown in Figure 4-8(b). Snapshots of some of the recognized activities and group micro-tasks, as well as detected formworks are shown in Figure 4-9. Table 4-7 shows the percentages of each micro-task from the detection time of one footing formwork to the detection time of the next one. In addition, it shows the average number of workers who worked during these periods. The results show that most of the idling happened during the first 45 minutes and the last 17 minutes. Since the video was recorded from 3 pm to 5 pm, the idling in the last 17 minutes was mostly related to workers preparing to leave the site after finishing the last formwork. However, the high percentage of idling in the first 45 minutes can indicate a potential issue, which will be investigated in Section 4.3.4.4. Regardless of the idling, workers spent most of their time from minute 30 on the "Formwork assembly" micro-task. However, as was discussed in Section 4.3.4.1, the relatively high percentages of "Walking" and "Transporting" activities during this micro-task could have reduced their productivity.



(a) The site at the beginning of the video (b) The site at the end of the video

Figure 4-8 Snapshots of the site on the 25th of September 2019 (case 1)

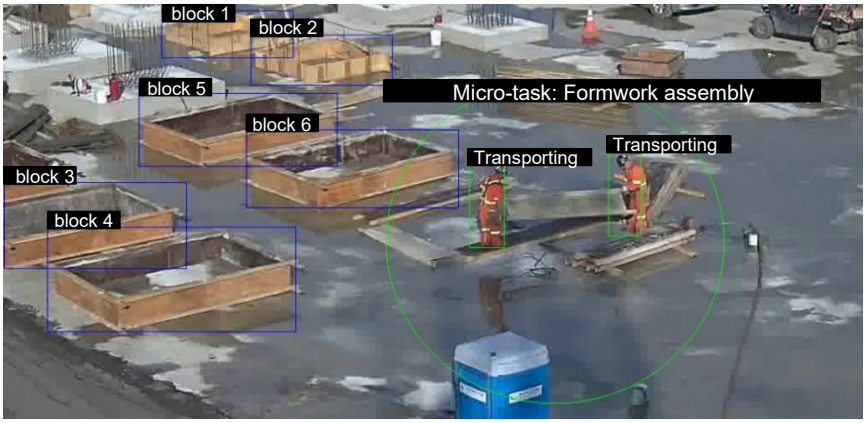


Figure 4-9 Snapshot of recognized activities of workers in a group, their group micro-task, and detected formworks

Table 4-7 Integrating progress and resource monitoring for better productivity analysis (case 1)

Time (min)	ID of the completed formwork	Duration percentages				Avg. no. workers
		Formwork assembly (%)	Adding footing reinforcement (%)	Idling (%)	Not defined (%)	
0 - 22	3	38.1	0.0	61.9	0.0	2.8
23 - 29	4	64.9	0.0	35.1	0.0	3.4
30 - 45	5	81.7	0.0	18.3	0.0	3.9
46 - 51	6	93.7	0.0	6.3	0.0	2.4
52 - 85	7	90.1	0.0	5.3	4.2	2.1
86 - 101	8	99.1	0.0	0.0	0.9	1.3
102 - 103	9	99.8	0.0	0.0	0.2	2.1
104 -120 (end)	-	72.5	0.0	25.7	1.8	1.7

Although the simplifying assumption, that there is only one group of workers working at a time, holds most of the time in case 1 video (explained in Section 4.2.4), parallel work by the same group of workers is observed. This is why the completion times presented in Table 4-7 have a large variance. Figure 4-10 shows the completion time of each formwork. For example, it took 22 minutes to assemble formwork 3, while formwork 4 was assembled in only 6 minutes. Manual observation of the video shows that workers first transported all the materials for both of these formworks, and then proceeded with hammering them together. In other words, workers built formwork 3 and formwork 4 together in 29 minutes, with the last 6 minutes being the time it takes to hammer the material together. Similarly, formworks 5 and 6 were built in parallel. This pattern is visible in Figure 4-10. Based on the above observation, hammering the material together takes between 5 to 6 minutes. Therefore, the transportation step took 18 minutes for formworks 3 and 4 (i.e., on average 9 minutes each), while it took 10 minutes for formworks 5 and 6 (i.e., 5 minutes each). Given more videos of a site, one can estimate the average time that each step takes, which can be used to identify parallel work and delays.

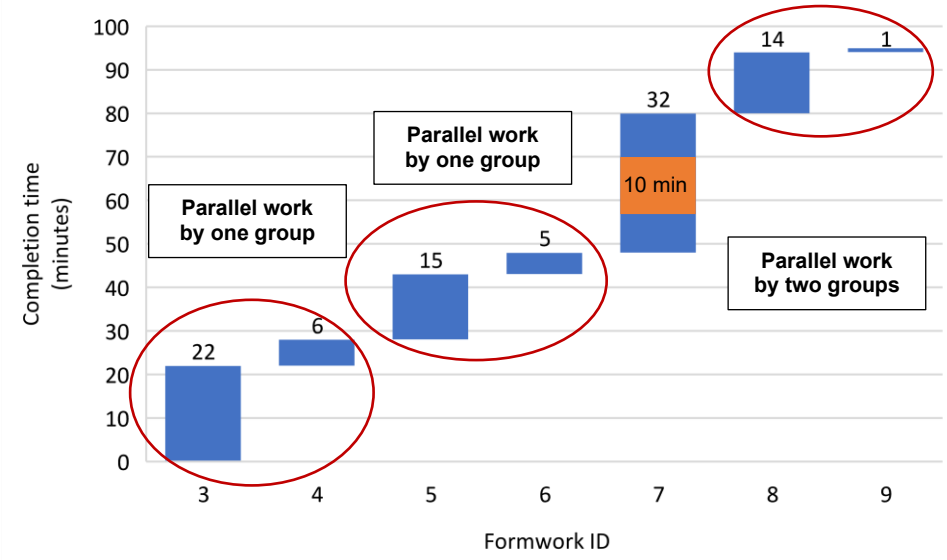


Figure 4-10 Completion time of footing formworks

As opposed to the first four formworks, formwork 7 was built alone (i.e., by a single worker and without parallel tasks). However, the involved workers left the camera's field of view in the middle of their work for about 10 minutes before completing this formwork. Finally, formworks 8 and 9 were built not only in parallel, but also by two separate groups of workers, which resulted in

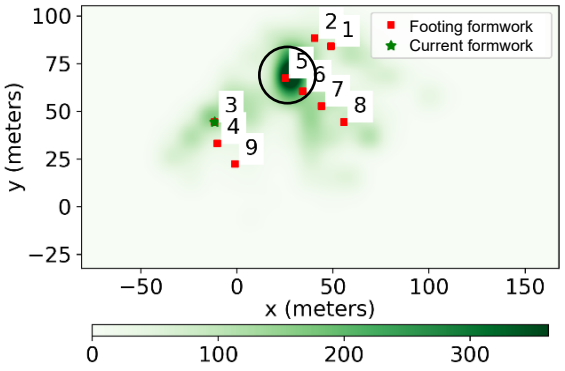
formwork 9 being completed only a minute later than formwork 8. Further discussion of potential future work to cover these different cases will be given in Section 4.4 .

4.3.4.4 Identifying the underlying reason behind the idling and high percentage of walking

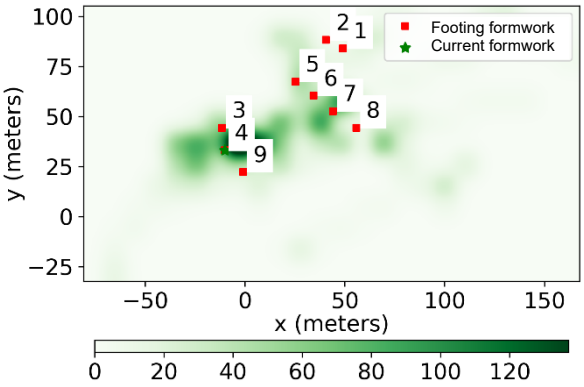
Table 4-7 showed that some of the 17.3% idling that was presented in Table 4-6 occurs during the first 45 minutes. In addition, based on the discussion in 4.9.1.1, there is a potential to improve the productivity of the "Formwork assembly" micro-task. In order to investigate these issues, the heatmaps of the "Walking" activity are generated in Figure 4-11 for each row of Table 4-7. For example, Figure 4-11(a) shows the location of workers during the first 22 minutes. This is the period when on average, 2.8 workers were working on footing formwork 3. The center locations of footing formworks are shown with red squares, and the footing formwork that is completed at the end of each period is shown with a green star at its center.

It is expected that the area around the footing formwork that is being built at any moment will have a higher density of walking activity compared to the rest of the site. Although this is usually the case, there are some additional areas, shown with circles, that have a high density of points in many cases. Manual observation of the video showed that this is usually where the wooden materials for the construction of footing formworks are kept, and workers constantly walked to this location to transport the materials. Optimal placement of the materials next to the planned location of each footing can help reduce the "Walking" and "Transporting" activities and improve the productivity of the "Formwork assembly" micro-task.

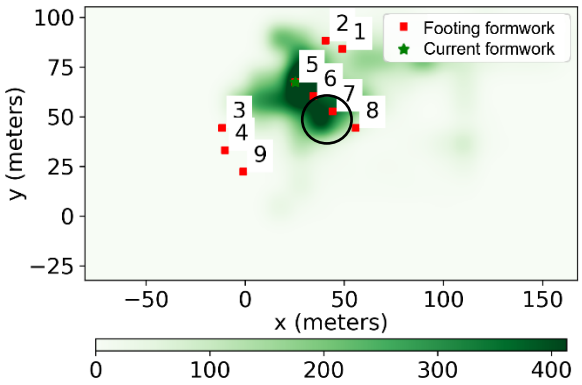
Some of the idling at the beginning of the video is caused by workers preparing to work on the site. Therefore, there are many instances when they are discussing the plan with each other and grabbing their tools. However, there is one particular event that can be prevented to reduce idling. As shown in Figure 4-11(a) and Figure 4-12(a), the laydown area is initially located near formwork 1. However, since additional formworks are going to be built at this location, it is later relocated (shown in Figure 4-11(c), Figure 4-12(b), and Figure 4-12(c)). This event caused workers to idle until the telehandler left the work zone. Therefore, non-optimal placement of materials not only reduces the productivity of workers by forcing them to walk and transport materials for longer distances, but also wastes time during relocation.



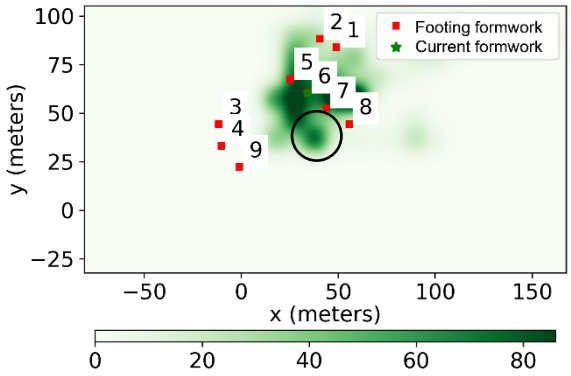
(a) Site from minute 0 to 22



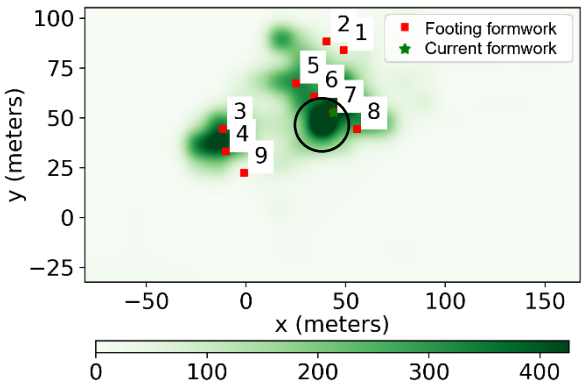
(b) Site from minute 23 to 29



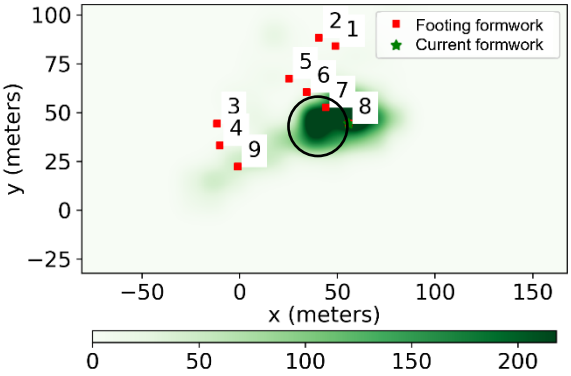
(c) Site from minute 30 to 45



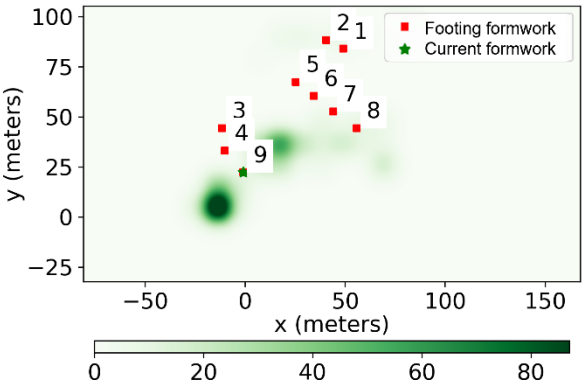
(d) Site from minute 46 to 51



(e) Site from minute 52 to 85

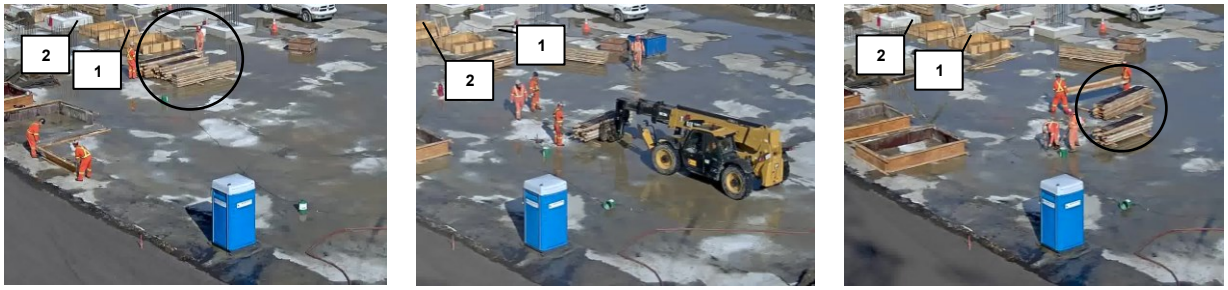


(f) Site from minute 86 to 101



(g) Site from minute 102 to 103

Figure 4-11 Walking heatmap per completed footing formwork (case 1)



(a) Initial (b) Relocation (c) Final

Figure 4-12 Snapshots of the laydown areas (case 1)

4.3.5 Case 2 – two micro-tasks

The productivity monitoring framework was also applied to a six-hour video of "Formwork assembly" and "Adding footing reinforcement" micro-tasks. The video also contains many instances of additional activities and micro-tasks that are not considered in this study. These instances will be discussed later in this section. Table 4-8 shows that "Walking", "Standing", and "Transporting" activities have the highest duration percentages 36.8%, 22.0%, and 14.6% respectively. Similar to case 1, the high percentages of walking and transporting materials can indicate idling, however, this should be investigated further by considering the corresponding micro-tasks. Table 4-9 shows that workers were in fact idling 30.8% of the time with a total of 75.7% of the idling times dedicated to the "Standing" and "Walking" activities. Additionally, 35.8% of the time, workers were busy with the "Formwork assembly" micro-task with 35.0% of "Walking", 17.5% of "Transporting", and 20.6% of value-adding "Hammering" activity.

As opposed to case 1, the material was initially placed near the workspace, as shown in Figure 4-13. However, they were used to only build a subset of formworks, and the rest of the materials were transported from a further location. Figure 4-14 shows the far placement of the material which forced workers to travel and transport them for longer distances, resulting in increasing the duration percentages of the "Walking" and "Transporting" activities. Moreover, the micro-task recognition method recognized that workers spent 21.3% of their time on "Adding footing reinforcement", which agrees with manual observation.

Table 4-8 Activity duration percentages in a six-hour video on the 30th of September 2019 (case 2)

Activity	Duration percentage (%)
Standing	22.0
Walking	36.8
Transporting	14.6
Measuring	1.1
Hammering	13.6
Drilling	4.1
Placing/Fixing rebars	5.9
Others	1.9

Table 4-9 Micro-task duration percentages in a six-hour video on the 30th of September 2019 (case 2)

Micro-task / Duration percentages	Formwork assembly (%)	Adding footing reinforcement (%)	Idling (%)	Not defined (%)	Gross total (%)
Total	35.8	21.3	30.8	12.1	100
Standing	23.0	7.9	28.2	25.8	
Walking	35.0	29.9	47.5	34.1	
Transporting	17.5	15.4	9.4	15.4	
Measuring	0.3	1.6	0.3	1.1	
Hammering	20.6	13.0	7.1	11.4	
Drilling	2.3	0.1	5.7	8.0	
Placing/Fixing rebars	0.2	31.7	0.0	0.4	
Others	1.1	0.4	1.8	3.8	
Gross total (%)	100	100	100	100	

The value-adding activity of the “Adding footing reinforcement” micro-task (i.e., “Placing/Fixing rebars”) has the highest duration percentage, with 31.7%, which shows the productivity of workers in the last hour of work. In contrast to case 1, a relatively high percentage of “Not defined” micro-tasks are recognized (i.e., 12.1%) which are mainly related to the preparation steps taken by workers, when switching between the two main micro-tasks (i.e., “Formwork assembly”, “Adding footing reinforcement”). Snapshots of some of these preparation steps are shown in Figure 4-15. Additionally, the high percentage of “Hammering” in the “Not defined” micro-task is related to the removal or placement of metallic bars that are placed around formworks to secure them. An example of this case is shown in Figure 4-16(a). Another example of these instances is related to workers using the hammer to make sure the wooden boards of the formworks are secure, as shown in Figure 4-16(b).

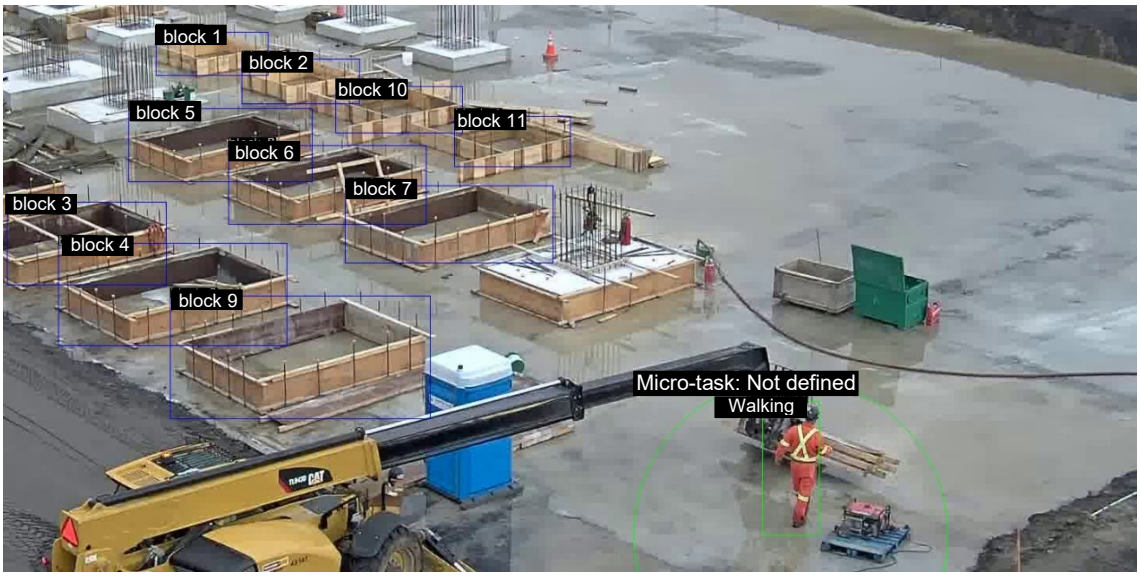


Figure 4-13 Placement of material near the workspace using a telehandler

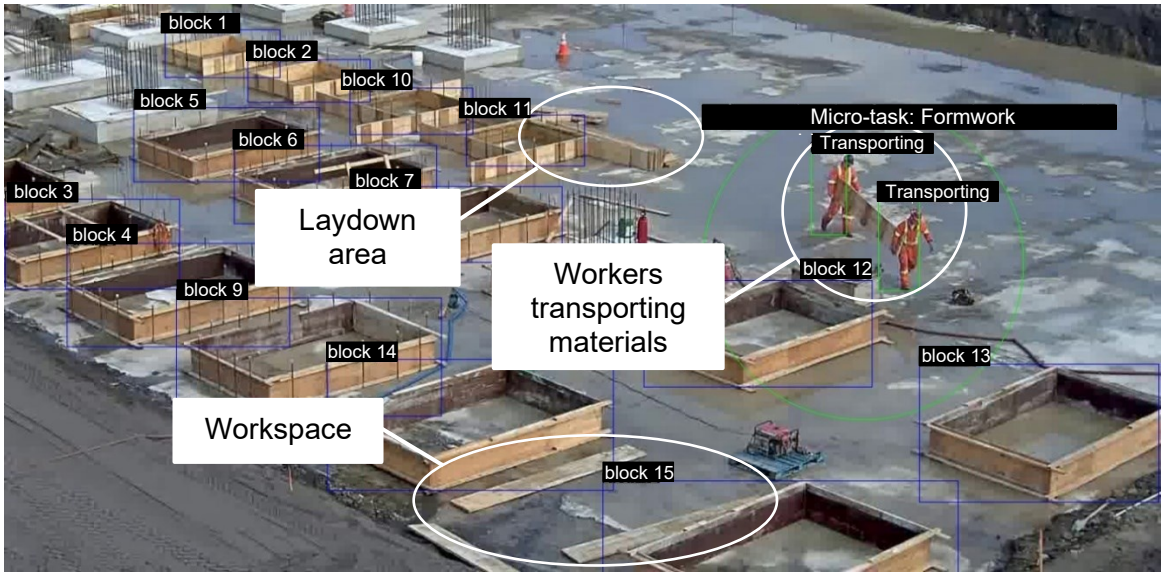
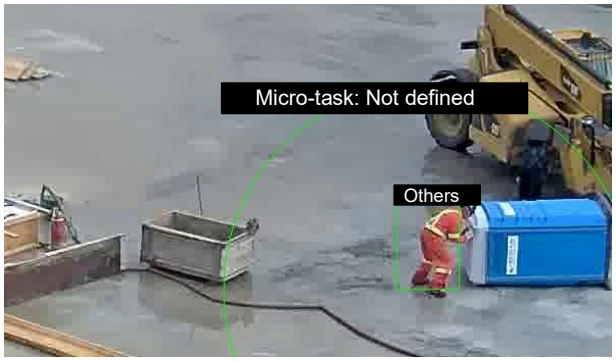
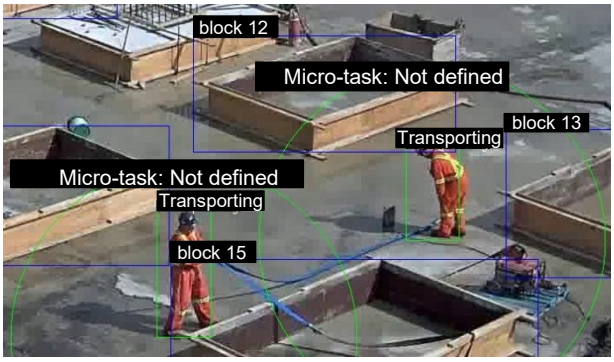


Figure 4-14 Placement of material far from the work zone

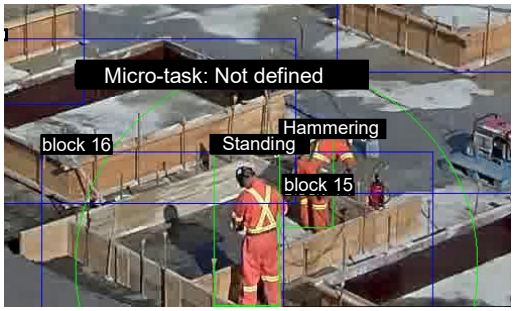


(a) Assistance in transportation

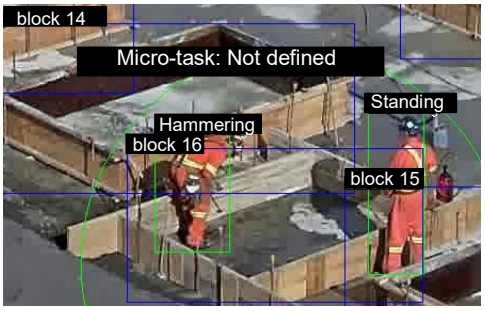


(b) Relocation of the extension cable

Figure 4-15 Snapshots of some of the "Not defined" micro-tasks (case 2)



(a) Securing the metallic support bars



(b) Securing the wooden boards

Figure 4-16 Snapshots of the "Hammering" activities for the "Not defined" micro-task

4.4 Discussion

The simplifying assumptions in Section 4.2.4 can be used for small constructions sites where only one group of workers is working most of the time. However, they do not hold for larger sites where multiple workers are working in parallel at different locations of the site. For these cases, temporal information should be considered together with spatial information. An example of this situation is shown in Figure 4-17, where there are two groups of workers who are working at separate locations in parallel. The first group (i.e., W_1, W_2) built product P_1 in Δt_1, and continued working on another product (i.e., P_3), which took Δt_3 to be built at a different location of the site. The second group of a single worker W_3 built P_2 in Δt_2, and continued working on another product which is not yet detected in Figure 4-17. This case is slightly more complex than Figure 4-5, as each group should be tracked now using available tracking methods (e.g., [66]), and should be associated with the detected products to calculate the completion times (i.e., Δt_1, Δt_2, Δt_3). To calculate Δt_1, the distance between the center location of P_1 at time t_1, and both work groups should first be computed. P_1 should then be associated with the closest group (i.e., W_1, W_2). Next, since P_1 is the first detected product, the time when the closest group to P_1 was first detected in the video should be found. Δt_1 can now be calculated as $t_1 - t_0$. Finding Δt_2 is slightly more complicated. After identifying the closest working group to P_2 at time t_2, this group (i.e., W_3) should be traced back to either the first time it was detected in the videos, or to the detection time of the previous product that was built by this group. Since P_2 is the first product built by W_3, Δt_2 can also be calculated as $t_2 - t_0$. On the other hand, since P_3 is the second product built by the first group (i.e., W_1, W_2), Δt_3 is calculated by subtracting the detection time of the last product which was built by this group (P_1, at t_1) from t_3. The process explained above can be used to calculate Δt_i for any number of products and any number of work groups. However, more complications can arise when the same group of workers starts working on multiple products in parallel as was seen in Section 4.3.4.3. These cases require detecting the start and end stages of the products (shown in Figure 4-4), in addition to the solution explained above.

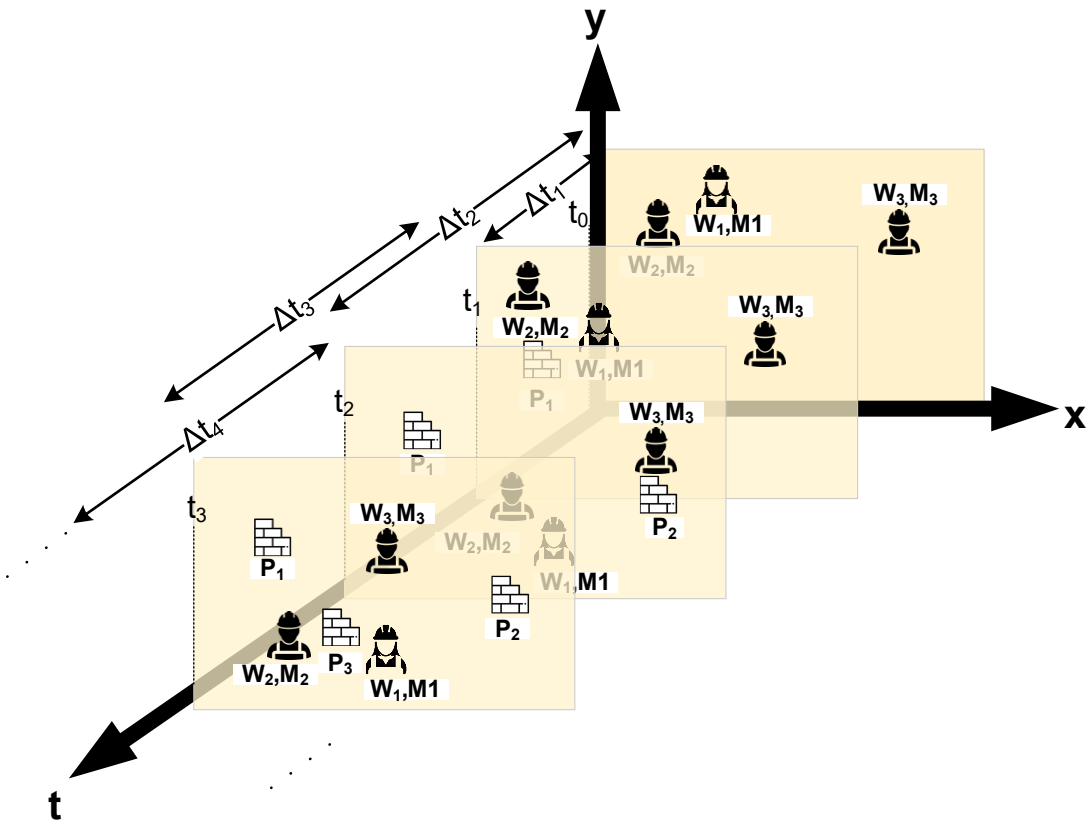


Figure 4-17 Parallel construction by multiple groups

4.5 Summary and Conclusions

Automatic productivity monitoring requires the extraction of data at different levels (e.g., activities, micro-tasks, and products) of construction operations. The previous CV-based method only focused on low-level activities. However, this research argues that this low-level information alone is not informative enough to identify idling, low productivity, and the underlying causes of low productivity. Therefore, this study tried to move one level higher than activities (taking place in few seconds), and recognize micro-tasks (taking place in few minutes), which can be further linked to the daily schedule. To this end, a novel micro-task recognition method is developed based on the combination of activities. In addition, as productivity is defined based on the relation between input and output, product detection is used to measure the progress of the site. Combining workers monitoring and progress monitoring gives the number and location of built products, their completion times, the average number of utilized workers, and the duration percentages of their

activities and micro-tasks, which can help project managers better estimate the completion time and budget.

The proposed framework is applied to a two-hour and a six-hour construction site videos, to recognize idling and group activities into higher level micro-tasks. A detailed analysis is performed based on a combination of activities, micro-tasks, and detected products, which proved that the proposed framework is promising.

Chapter 5: Conclusions and Future Work

5.1 Summary of Research

This research investigated the applicability of CV-based activity recognition and object detection for automatic productivity monitoring of construction workers. The related literature was reviewed to identify the limitations, search gaps, and potential solutions. A CV-based productivity monitoring framework was proposed to address some of the identified limitations.

In Chapter 2, some of the commonly used productivity measuring techniques were first reviewed and the challenges of applying them were discussed. Then, the literature in CV-based activity classification, as a possible solution for productivity monitoring automation was reviewed. Next, previous CV-based workers' productivity monitoring solutions in the construction domain were presented and their limitations and gaps were discussed. Finally, the literature in spatiotemporal activity recognition was reviewed and a suitable method was selected to address some of the limitations of the previous activity recognition method used in the construction domain (referred to as the three-stage method).

In Chapter 3, the steps for training and validation of the proposed CV-based spatiotemporal activity recognition method were first presented. A video dataset including six common activities of construction workers (i.e., "Standing", "Walking", "Transporting", "Drilling", "Hammering", and "Placing/fixing rebars") was manually prepared and annotated. A semi-automatic annotation framework was proposed to facilitate the per-frame and per-worker annotation of the dataset. The custom dataset was used to train and validate the proposed method called YOWO and an improved version of it was introduced in this research (YOWO53) for spatiotemporal activity recognition of construction workers. The method is jointly optimized for per-frame and per-worker detection and activity classification and can be applied on long untrimmed surveillance videos of multiple construction workers each performing different activities and switching between them continuously. YOWO and YOWO53 were extensively compared together and to the previous three-stage method.

Chapter 4 tried to move one level higher than activities recognized in Chapter 3, and recognize micro-tasks, which can be further linked to the daily schedule. A novel micro-task recognition

method was developed based on the combination of activities. In addition, as productivity is defined based on the relation between input and output, product detection was used to measure the progress of the site. Combining workers monitoring and progress monitoring gives the number and location of built products, their completion times, the average number of workers, and the duration percentage of their activities and micro-tasks, which can help project managers better estimate the completion time, and required resources.

5.2 Research Contributions and Conclusions

This research made the following contributions to the body of knowledge:

(1) A CV-based spatiotemporal activity recognition method (i.e., YOWO) is used and further improved (i.e., YOWO53) to address the limitations of the previously used activity recognition method in the construction domain (i.e., three-stage method).

(2) A sensitivity analysis is conducted to compare the proposed and the previous methods in terms of detection, classification, and overall activity recognition performance, as well as the speed, size, and computational complexity. With regards to this contribution, the following conclusions can be drawn:

- A fully optimized and single-stage spatiotemporal activity recognition method such as YOWO and YOWO53, that jointly detects workers and classifies their activities, is superior to the previously widely used three-stage method, which required separate optimization of three different modules. The joint methods improved the activity classification accuracy by at least 15%.
- Replacing the 2D backbone of YOWO (Darknet-19) with Darknet-53, and decreasing the size of its receptive field (resulting in YOWO53), improved the detection performance of the model by at least 2% for small objects such as construction workers in far-field surveillance videos.
- As the size of the input frame reduces, the speed increases and the superiority of YOWO53 over YOWO in terms of detection performance becomes more tangible.

- YOWO and YOWO53 require more GPU memory and are much slower than the three-stage method. However, real-time speed is not required for productivity monitoring, and better detection and activity classification is preferred over speed.

(3) A novel CV-based method was proposed to recognize construction workers' higher-level micro-tasks from their low-level activities. With regards to this contribution, the following conclusions can be drawn:

- Automatic productivity monitoring requires the extraction of data at different levels (e.g., activities, micro-tasks, and products) of construction operations.
- Low-level information alone is not informative enough to identify idling, low productivity, and their underlying reasons.

(4) A detailed analysis was conducted on two case study videos (two and six hours) from a real construction site, to show that the proposed method can be used for the following tasks:

- Computation of the percentages of workers' time spent on different activities and micro-tasks, as well as different activities per micro-task, which help identify idling and low productivity and their underlying reasons. These numbers can also be used as the input to different productivity measurement methods.
- Measuring the progress of the site with product detection and combining it with resource monitoring by counting the number of the built products, calculating their completion times, the average number of utilized resources (i.e., workers), and their micro-tasks.

5.3 Limitations and Future Work

Despite the contributions of this work, there are still some limitations to be addressed in the future as follows. Each limitation is followed by a possible solution and future work.

(1) Like all supervised DL-based approaches, the need for a large dataset is one of the biggest shortcomings of this method in practice.

- In addition to data sharing, one way to work around this issue is by using self-supervised methods [72, 73], which are gaining a lot of interest among artificial intelligence researchers.

(2) The relatively small size of the activity recognition dataset hindered the following experiments that can be conducted in the future given more data:
    - Using a separate validation and test set for a more accurate performance evaluation of the model and fine-tuning of hyperparameters.
    - Fine-tuning further layers of the pretrained 2D and 3D backbones.

(3) The method proposed in this study requires high-resolution frames. When workers are far from the camera, activities that are done with different tools, but have the same pose and mobility aspects, become hard to distinguish from one another. The fact that workers are not always facing the camera makes activity recognition even harder. An example of this situation is shown in Figure 5-1, where the hammering activity is recognized correctly when the hammer is visible in Figure 5-1(a) and recognized wrongly as measuring when the hammer is not visible in Figure 5-1(b).
    - Using multiple cameras can help with some of these cases.
    - The daily project schedule can be used to remove unexpected and wrongly recognized activities, which can help with activity recognition of the workers who are not facing the camera or the highly similar activities.

(4) The micro-tasks are recognized for groups of workers due to the challenges of tracking workers. However, workers' interactions in groups are not leveraged for activity recognition. Even if workers are interacting with each other, activities are recognized for each worker separately.
    - As the tracking methods improve, future works can try to track each worker to recognize their micro-tasks individually and estimate their productivity for comparison.
    - In addition, workers' interactions with each other (as well as with detected objects) can be leveraged by using graph convolutional neural networks to improve activity recognition performance. [74]

(5) The simplifying assumption of non-parallel work by multiple groups on the products, and non-parallel work on multiple products by a single group is used in the case study of this research.

In addition, only the ending of the product of one of the micro-tasks (i.e., completed footing formworks as the output of formwork assembly micro-tasks) was detected.

- Introducing more stages of different products (e.g., Figure 4-4), including the initial and completed stages of products of different micro-tasks.
- Tracking workers or work groups and associating them with products based on their spatial location.

(6) The micro-tasks and their locations can be used in the future to automatically create the as-built 4D simulation, which can be compared with the as-planned 4D simulation to identify deviations and potential solutions.

- The resource and progress monitoring methods proposed in this thesis provide detailed information about the location of workers and products, the status of products, and the activities, and micro-tasks of workers at any second. These data can be imported to different simulation software or game engines to automatically create detailed 4D as-built simulations.

(7) In addition, the proposed heuristic-based micro-task recognition method is not able to correctly recognize instances of micro-tasks with undefined patterns, resulting in "Not defined" micro-tasks that are not necessarily unproductive.

- In the future, more micro-tasks should be defined to include these instances.

(a) Visible hammer (b) Hammer not visible

Figure 5-1 An instance of hammering confused with measuring

(8) From the three main modules of the proposed productivity monitoring framework, only activity recognition and product detection modules were evaluated. The performance of the micro-task recognition module was only evaluated by visually comparing the results to the videos. In

addition, the data extracted by the proposed framework was not used to calculate productivity rates for comparison with the manual productivity measuring approach.

- In the future, an annotated micro-task dataset should be prepared to evaluate the accuracy of the micro-task recognition method.
- The resource and progress data should be used to calculate productivity rates and compared to the productivity rates achieved with manual data collection in terms of speed and accuracy.

(9) The thresholds α and β used in the micro-task recognition method were chosen based on manual trial and error and the percentage of unmatched micro-tasks.

- Given training micro-task recognition data, α and β can be optimized for the lowest error among all micro-tasks.

(10) Although the three main modules of the proposed framework (i.e., activity recognition, micro-task recognition, and product detection) were automated, the analysis of this data was done manually.

- In the future, maximum allowed percentages of walking and standing within each micro-task, as well as maximum percentage of idling can be chosen by experienced project supervisors and foremen to automatically identify potential issues. The framework can then automatically retract the candidate video segments for further analysis.

REFERENCES

[1] S. P. Dozzi, S. M. AbouRizk, National Research Council Canada, and Institute for Research in Construction (Canada), *Productivity in construction*. Ottawa: Institute for Research in Construction, National Research Council, 1995.

[2] Z. U. Khan, "Modeling and parameter ranking of construction labor productivity," masters, Concordia University, 2005. Accessed: Feb. 07, 2022. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/8615/

[3] C. Chen, Z. Zhu, and A. Hammad, "Automated excavators activity recognition and productivity analysis from construction site surveillance videos," *Automation in Construction*, vol. 110, p. 103045, Feb. 2020, doi: 10.1016/j.autcon.2019.103045.

[4] X. Luo, H. Li, D. Cao, F. Dai, J. Seo, and S. Lee, "Recognizing Diverse Construction Activities in Site Images via Relevance Networks of Construction-Related Objects Detected by Convolutional Neural Networks," *J. Comput. Civ. Eng.*, vol. 32, no. 3, p. 04018012, May 2018, doi: 10.1061/(ASCE)CP.1943-5487.0000756.

[5] X. Luo, H. Li, D. Cao, Y. Yu, X. Yang, and T. Huang, "Towards efficient and objective work sampling: Recognizing workers' activities in site surveillance videos with two-stream convolutional networks," *Automation in Construction*, vol. 94, pp. 360–370, Oct. 2018, doi: 10.1016/j.autcon.2018.07.011.

[6] X. Luo, H. Li, H. Wang, Z. Wu, F. Dai, and D. Cao, "Vision-based detection and visualization of dynamic workspaces," *Automation in Construction*, vol. 104, pp. 1–13, Aug. 2019, doi: 10.1016/j.autcon.2019.04.001.

[7] X. Luo, H. Li, Y. Yu, C. Zhou, and D. Cao, "Combining deep features and activity context to improve recognition of activities of workers in groups," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 9, pp. 965–978, Sep. 2020, doi: 10.1111/mice.12538.

[8] D. Roberts and M. Golparvar-Fard, "End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level," *Automation in Construction*, vol. 105, p. 102811, Sep. 2019, doi: 10.1016/j.autcon.2019.04.006.

[9] W. Kay *et al.*, "The Kinetics Human Action Video Dataset," *arXiv:1705.06950 [cs]*, May 2017, Accessed: Mar. 28, 2020. [Online]. Available: http://arxiv.org/abs/1705.06950

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," 2011.

[11] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[12] C. Gu, "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6047–6056.

[13] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 961–970. doi: 10.1109/CVPR.2015.7298698.

[14] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding," *arXiv:1604.01753 [cs]*, Jul. 2016, Accessed: Dec. 23, 2020. [Online]. Available: http://arxiv.org/abs/1604.01753

[15] H. Idrees *et al.*, "The THUMOS challenge on action recognition for videos 'in the wild,'" *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, Feb. 2017, doi: 10.1016/j.cviu.2016.10.018.
[16] D. Moltisanti, S. Fidler, and D. Damen, "Action Recognition From Single Timestamp Supervision in Untrimmed Videos," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 9907–9916. doi: 10.1109/CVPR.2019.01015.
[17] G. Yao, T. Lei, X. Liu, and P. Jiang, "Temporal Action Detection in Untrimmed Videos from Fine to Coarse Granularity," *Applied Sciences*, vol. 8, no. 10, 2018, doi: 10.3390/app8101924.
[18] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos," *Int J Comput Vis*, vol. 126, no. 2–4, pp. 375–389, Apr. 2018, doi: 10.1007/s11263-017-1013-y.
[19] M. Everingham *et al.*, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010, doi: 10.1007/s11263-009-0275-4Everingham.
[20] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," *arXiv:1405.0312 [cs]*, Feb. 2015, Accessed: Jun. 07, 2021. [Online]. Available: http://arxiv.org/abs/1405.0312
[21] D. W. Halpin and L. S. Riggs, "Planning and Analysis of Construction Operations," p. 1.
[22] "General purpose simulation with stroboscope | Proceedings of the 26th conference on Winter simulation." https://dlnext.acm.org/doi/abs/10.5555/193201.194688 (accessed Sep. 23, 2021).
[23] J. Gong and C. H. Caldas, "Computer Vision-Based Video Interpretation Model for Automated Productivity Analysis of Construction Operations," *J. Comput. Civ. Eng.*, vol. 24, no. 3, pp. 252–263, May 2010, doi: 10.1061/(ASCE)CP.1943-5487.0000027.
[24] J. Gong and C. H. Caldas, "An object recognition, tracking, and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations," *Automation in Construction*, vol. 20, no. 8, pp. 1211–1226, Dec. 2011, doi: 10.1016/j.autcon.2011.05.005.
[25] G. Torabi, A. Hammad, and N. Bouguila, "2D and 3D CNN Based Simultaneous Detection and Activity Classification of Construction Workers," *J. Comput. Civ. Eng.*.
[26] 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," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 652–663, Oct. 2013, doi: 10.1016/j.aei.2013.09.001.
[27] J. Cai, Y. Zhang, and H. Cai, "Two-step long short-term memory method for identifying construction activities through positional and attentional cues," *Automation in Construction*, vol. 106, p. 102886, Oct. 2019, doi: 10.1016/j.autcon.2019.102886.
[28] S. Jung, J. Jeoung, H. Kang, and T. Hong, "3D convolutional neural network-based one-stage model for real-time action detection in video of construction equipment," *Computer-Aided Civil and Infrastructure Engineering*, p. mice.12695, Jun. 2021, doi: 10.1111/mice.12695.
[29] J. Kim and S. 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, Aug. 2019, doi: 10.1016/j.autcon.2019.03.025.

[30] J. Yang, M.-W. Park, P. A. Vela, and M. Golparvar-Fard, “Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future,” *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 211–224, Apr. 2015, doi: 10.1016/j.aei.2015.01.011.
[31] M. Golparvar-Fard, F. Peña-Mora, and S. Savarese, “Integrated Sequential As-Built and As-Planned Representation with D4AR Tools in Support of Decision-Making Tasks in the AEC/FM Industry,” *Journal of Construction Engineering and Management*, vol. 137, no. 12, pp. 1099–1116, Dec. 2011, doi: 10.1061/(ASCE)CO.1943-7862.0000371.
[32] K. K. Han and M. Golparvar-Fard, “Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs,” *Automation in Construction*, vol. 53, pp. 44–57, May 2015, doi: 10.1016/j.autcon.2015.02.007.
[33] H. Son, C. Kim, and C. Kim, “Automated Color Model–Based Concrete Detection in Construction-Site Images by Using Machine Learning Algorithms,” *Journal of Computing in Civil Engineering*, vol. 26, no. 3, pp. 421–433, May 2012, doi: 10.1061/(ASCE)CP.1943-5487.0000141.
[34] M. G. Fard and F. Peña-Mora, “Application of Visualization Techniques for Construction Progress Monitoring,” pp. 216–223, Apr. 2012, doi: 10.1061/40937(261)27.
[35] M. Golparvar-Fard, F. Pea-Mora, C. A. Arboleda, and S. Lee, “Visualization of construction progress monitoring with 4D simulation model overlaid on time-lapsed photographs,” *Journal of Computing in Civil Engineering*, vol. 23, no. 6, pp. 391–404, 2009, doi: 10.1061/(ASCE)0887-3801(2009)23:6(391).
[36] Y. M. Ibrahim, T. C. Lukins, X. Zhang, E. Trucco, and A. P. Kaka, “Towards automated progress assessment of workpackage components in construction projects using computer vision,” *Advanced Engineering Informatics*, vol. 23, no. 1, pp. 93–103, Jan. 2009, doi: 10.1016/j.aei.2008.07.002.
[37] X. Zhang *et al.*, “Automating progress measurement of construction projects,” *Automation in Construction*, vol. 18, no. 3, pp. 294–301, May 2009, doi: 10.1016/j.autcon.2008.09.004.
[38] R. Girdhar, J. Joao Carreira, C. Doersch, and A. Zisserman, “Video Action Transformer Network,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 244–253. doi: 10.1109/CVPR.2019.00033.
[39] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, “Action Tubelet Detector for Spatio-Temporal Action Localization,” *arXiv:1705.01861 [cs]*, Aug. 2017, Accessed: Jul. 28, 2020. [Online]. Available: http://arxiv.org/abs/1705.01861
[40] O. Köpüklü, X. Wei, and G. Rigoll, “You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization,” *arXiv:1911.06644 [cs]*, Jun. 2020, Accessed: Jul. 16, 2020. [Online]. Available: http://arxiv.org/abs/1911.06644
[41] J. Pan, S. Chen, Z. Shou, J. Shao, and H. Li, “Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization,” *arXiv:2006.07976 [cs, eess]*, Jul. 2020, Accessed: Jul. 22, 2020. [Online]. Available: http://arxiv.org/abs/2006.07976
[42] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. Davis, and J. Kautz, “STEP: Spatio-Temporal Progressive Learning for Video Action Detection,” *arXiv:1904.09288 [cs]*, Apr. 2019, Accessed: Jul. 22, 2020. [Online]. Available: http://arxiv.org/abs/1904.09288
[43] M. E. Manser, “Productivity Measures for Retail Trade: Data and Issues Productivity Measures for Retail Trade,” *Monthly Lab. Rev.*, vol. 128, no. 7, pp. 30–38, 2005.

[44] H. R. Thomas and D. F. Kramer, *The manual of construction productivity measurement and performance evaluation*. Bureau of Engineering Research, University of Texas at Austin, 1988.
[45] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, Jun. 2013, doi: 10.1016/j.cviu.2013.01.013.
[46] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A Short Note about Kinetics-600," *arXiv:1808.01340 [cs]*, Aug. 2018, Accessed: Jan. 27, 2022. [Online]. Available: http://arxiv.org/abs/1808.01340
[47] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A Short Note on the Kinetics-700 Human Action Dataset," *arXiv:1907.06987 [cs]*, Jul. 2019, Accessed: Feb. 28, 2021. [Online]. Available: http://arxiv.org/abs/1907.06987
[48] A. Li, M. Thotakuri, D. A. Ross, J. Carreira, A. Vostrikov, and A. Zisserman, "The AVA-Kinetics Localized Human Actions Video Dataset," *arXiv:2005.00214 [cs, eess]*, May 2020, Accessed: Feb. 28, 2021. [Online]. Available: http://arxiv.org/abs/2005.00214
[49] J. Donahue, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.
[50] L. Wang *et al.*, "Temporal Segment Networks for Action Recognition in Videos," *arXiv:1705.02953 [cs]*, May 2017, Accessed: Jan. 30, 2020. [Online]. Available: http://arxiv.org/abs/1705.02953
[51] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?," *arXiv:1711.09577 [cs]*, Apr. 2018, Accessed: Jan. 30, 2020. [Online]. Available: http://arxiv.org/abs/1711.09577
[52] G. Varol, I. Laptev, and C. Schmid, "Long-Term Temporal Convolutions for Action Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018, doi: 10.1109/TPAMI.2017.2712608.
[53] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, vol. 1, p. I-511-I–518. doi: 10.1109/CVPR.2001.990517.
[54] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767 [cs]*, Apr. 2018, Accessed: Jun. 08, 2021. [Online]. Available: http://arxiv.org/abs/1804.02767
[55] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, Sep. 2016, doi: 10.1109/ICIP.2016.7533003.
[56] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999, doi: 10.1145/304181.304187.
[57] J. Li, G. Zhou, D. Li, M. Zhang, and X. Zhao, "Recognizing workers' construction activities on a reinforcement processing area through the position relationship of objects detected by faster R-CNN," *Engineering, Construction and Architectural Management*, vol. ahead-of-print, no. ahead-of-print, Jan. 2022, doi: 10.1108/ECAM-04-2021-0312.
[58] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 21–37.

[59] L. Song, S. Zhang, G. Yu, and H. Sun, "TACNet: Transition-Aware Context Network for Spatio-Temporal Action Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, 2019, pp. 11979–11987. doi: 10.1109/CVPR.2019.01226.
[60] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
[61] O. Köpüklü, N. Kose, A. Gunduz, and G. Rigoll, "Resource Efficient 3D Convolutional Neural Networks," *arXiv:1904.02422 [cs]*, Sep. 2019, Accessed: Jun. 27, 2020. [Online]. Available: http://arxiv.org/abs/1904.02422
[62] H. Le and A. Borji, "What are the Receptive, Effective Receptive, and Projective Fields of Neurons in Convolutional Neural Networks?," *arXiv:1705.07049 [cs]*, Apr. 2018, Accessed: Sep. 14, 2021. [Online]. Available: http://arxiv.org/abs/1705.07049
[63] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, 1973, doi: 10.1109/PROC.1973.9030.
[64] J. Redmon, "Darknet: Open Source Neural Networks in C," *Pjreddie.com*, 2021. https://pjreddie.com/darknet/ (accessed Jan. 21, 2021).
[65] T. LabelImg, "Git code." 2015. [Online]. Available: https://github.com/tzutalin/labelImg
[66] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, 2017, pp. 3645–3649.
[67] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 1106–1112. doi: 10.1109/CVPR.1997.609468.
[68] C. Chen, Z. Zhu, A. Hammad, and M. Akbarzadeh, "Automatic Identification of Idling Reasons in Excavation Operations Based on Excavator–Truck Relationships," *J. Comput. Civ. Eng.*, vol. 35, no. 5, p. 04021015, Sep. 2021, doi: 10.1061/(ASCE)CP.1943-5487.0000981.
[69] B. Kulis and M. I. Jordan, "Revisiting k-means: New Algorithms via Bayesian Nonparametrics," p. 8.
[70] A. Zang, D. Lucio, and L. Velho, "QtCalib Software." Rio de Janeiro.
[71] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, Aug. 1987, doi: 10.1109/JRA.1987.1087109.
[72] K. Kahatapitiya, Z. Ren, H. Li, Z. Wu, and M. S. Ryoo, "Self-supervised Pretraining with Classification Labels for Temporal Activity Detection," *arXiv:2111.13675 [cs]*, Nov. 2021, Accessed: Dec. 16, 2021. [Online]. Available: http://arxiv.org/abs/2111.13675
[73] L. Tao, X. Wang, and T. Yamasaki, "Self-supervised Video Representation Learning Using Inter-intra Contrastive Framework," in *Proceedings of the 28th ACM International Conference on Multimedia*, New York, NY, USA, Oct. 2020, pp. 2193–2201. doi: 10.1145/3394171.3413694.
[74] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning Actor Relation Graphs for Group Activity Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 9956–9966. doi: 10.1109/CVPR.2019.01020.

Appendix A: List of Publications

Torabi, G., Hammad, A., & Bouguila, N. (2022). Two-Dimensional and Three-Dimensional CNN-Based Simultaneous Detection and Activity Classification of Construction Workers. *Journal of Computing in Civil Engineering*, *36*(4), 04022009. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001024

Huang, Y., Hammad, A., Torabi, G., Ghelmani, A., & Guevremont, M. (2021). Towards Near Real-time Digital Twins of Construction Sites: Developing High-LOD 4D Simulation Based on Computer Vision and RTLS. *Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC)*, 248–255. https://doi.org/10.22260/ISARC2021/0036

Torabi, G., Hammad, A., & Bouguila, N. (2021). Joint Detection and Activity Recognition of Construction Workers Using Convolutional Neural Networks. *Proceedings of the 2021 European Conference on Computing in Construction*, *2*, 212–219. https://doi.org/10.35490/ec3.2021.197