ITcon

www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

SMART CONSTRUCTION SCHEDULING MONITORING USING YOLOV3-BASED ACTIVITY DETECTION AND CLASSIFICATION

SUBMITTED: May 2021
REVISED: September 2021
PUBLISHED: March 2022
EDITOR: Žiga Turk
DOI: 10.36680/j.itcon.2022.012

Shubham Bhokare
School of Electrical and Computer Engineering, West Lafayette, IN, United States, Purdue University
sbhokare@purdue.edu

Lakshya Goyal
School of Electrical and Computer Engineering, West Lafayette, IN, United States, Purdue University
lgoyal@purdue.edu

Ran Ren
School of Construction Management Technology, West Lafayette, IN, United States, Purdue University
ren153@purdue.edu

Jiansong Zhang, Ph.D.
School of Construction Management Technology, West Lafayette, IN, United States, Purdue University
Zhan3062@purdue.edu

SUMMARY: *Increasing efficiency and adhering to a schedule are prominent issues faced by many construction projects. Identifying areas where productivity is low would automatically be a helpful tool for managers. This research aims to analyze and compare the efficiency and accuracy of different computer-vision based activity recognition algorithms that are used on construction sites. The authors then propose a method which involves the use of YOLOv3 to perform activity recognition on construction sites and compare the accuracy of our method to existing algorithms. The algorithms for comparison are selected on the basis that: (1) they incorporate various state-of-the-art activity recognition techniques, such as bounding-box predictions and skeleton-models; and (2) they are relatively recent implementations. The authors trained the model using a data-base consisting of 4 activities with frames from 20 videos for each. The dataset was created by extracting frames from the videos and labelling the activities that are taking place in each video. The authors then use the aforementioned activity classification method to propose a smart schedule monitoring system that automatically updates start and finish times of individual activity conducted in a construction project based on the activities that are detected. This computer-vision based approach to provide automatic and real-time updates to the construction schedule is expected to improve worker productivity and shorten construction project timelines.*

KEYWORDS: *You only look once (YOLO)v3, Activity Recognition, Construction Schedule Monitoring, Building Information Modeling (BIM).*

REFERENCE: *Shubham Bhokare, Lakshya Goyal, Ran Ren, Jiansong Zhang (2022). Smart construction scheduling monitoring using YOLOv3-based activity detection and classification. Journal of Information Technology in Construction (ITcon), Vol. 27, pg. 240-252, DOI: 10.36680/j.itcon.2022.012*

1 INTRODUCTION

One of the biggest challenges faced by construction projects is staying on schedule. Construction projects are often behind schedule. A study in Nepal found that the mean actual duration to complete about 50% of the construction projects was 24.7 months, whereas the planned average duration was 15.2 months for 75% of the projects (Mishra and Bhandari 2018). These types of delays cause a project to be severely behind schedule and lead to high costs and negative economic impacts. There are many aspects of a project that can cause these delays, such as unrealistic contract durations, the contractor's financial problems, poor labour productivity, project size, the delay of material delivery to site, design changes, and weather, among others (Mpofu et al. 2017; Abbasi et al. 2020; Agyekum-Mensah and Knight 2017; Durdyev et al. 2017; Alwi and Hampson 2003). In terms of a successful project, although there are many factors from human aspect that could contribute to a project, such as trades' skill, and distribution of labor, one of the largest contribution comes from the human aspect is labour productivity of the construction workers on the site (Mpofu et al. 2017). Not only that, this labour cost could account for 33-50% of total project costs (Gouett et al. 2011). Therefore this is the area this research focuses upon. Being able to improve productivity and efficiency in this area can help construction projects save a lot of time and money. Worker productivity is difficult to track, measure and record, but that is the first step towards finding ways to improve it. The authors argue that firstly there needs to be a way to identify where productivity is low, and only then can we start finding ways to improve it. More specifically, people need to know where on a construction site the delays start to build up. Many construction sites currently rely on the foreman to collect information regarding the progress on the site and what tasks are being done (Yang et al. 2016). This adds additional duties for them and takes time away from them doing more critical tasks, such as quality control and safety inspections. This is where technology can kick in. With the advancement in computer vision technologies and the wide availability of implementations, we can find a better way to track worker activities.

The goal of this project is to build a smart scheduling system for real-time activity tracking of construction site progress. The authors aim to design and implement a program that uses computer vision to detect activities taking place on the construction site to update the schedule and timeline of progress automatically. Construction sites right now often lack implementations of modern technology (Blanco et al. 2017). With the rise of 4D Building Information Modelling (BIM), the authors wanted to test cutting-edge computer vision methods in providing valuable information for real-time schedule monitoring.

The paper is organized as follows. Firstly, the authors review the state-of-the-art methods in literature to perform recognition of activities performed by workers on a construction site and explain the methodology behind them and their performance measures on relevant datasets. Secondly, the authors give an in-depth overview of the proposed activity recognition solution including the algorithm, the training and testing procedures, and an explanation of the solution's performance measures. This section also talks about the dataset assembled for the purpose of training and testing and how the results of activity recognition compare to other methods used for similar purposes. Thirdly, the authors explain the details of the proposed smart scheduling system and how it incorporates the aforementioned activity recognition method to update project timelines in real-time.

2 RESEARCH BACKGROUND

Methods involving activity recognition, and the tracking of workers and construction equipment have been gaining a lot of popularity in recent years in the construction industry. This rise, both in popularity and usage has coincided with the exponentially growing advances in real-time localization and/or identification technologies including Global Positioning System (GPS) and Radio Frequency Identification (RFID) (Gong et al. 2011) as well as technologies that involve the use of computer vision and Convolutional Neural Networks (CNN) (Luo et al. 2018a).

Most of the real-time localization and/or identification techniques involve the use of different types of sensors. Some of the commonly used techniques include the usage of smartphones as a means of recognizing the activities performed by the construction workers (Akhavian and Behzadan 2016; Akhavian et al. 2015). Other types of sensors used for the purpose of activity classification include accelerometers (Joshua and Varghese 2011), wearable devices (Kao et al. 2009; Akhavian and Behzadan 2015a; Akhavian and Behzadan 2018) and audio signal processors (Cheng et al. 2016). The data collected from these sensors is often used as training data for machine learning classifiers (Akhavian and Behzadan 2015b) which are in turn used to classify activities in order to improve worker productivity and safety standards of a construction site (Joshua and Varghese 2011). With rapid developments in the field of computer vision and machine learning, aided by the increasing resolutions of cameras and larger database and storage capacities (Luo et al. 2018a), techniques of classification that involve the use of these concepts have become increasingly common.

2.1 Current activity classifiers used for construction sites

2.1.1 Bag-of-video-feature-words and Bayesian network models

The Bag-of-Video-Feature-Words (BoF) method (Gong et al. 2011) has been one of the most successful methods proposed for the purpose of activity classification on construction sites (Yang et al. 2016). This method consists of two main stages: the learning stage and the recognition stage. The learning stage of the method involves modeling image sequences using an operator proposed by Laptev (Laptev 2005), conducting feature detection and representation, and performing vector quantization using k-means clustering for generating a Codebook. This stage also involves learning the model using two different approaches, namely, Naïve Bayesian classifier and probabilistic Latent Semantic Analysis (pLSA). The recognition stage involves applying the learned action model to classify test data into activities performed by workers. This method was tested on two video datasets, a dataset of backhoe actions (3 activities, 150 videos) and a dataset of worker actions (5 activities, 300 videos). The method managed to achieve an average accuracy of activity classification in the range of 73.6%-79% and was able to perform activity recognition on a new video in about 0.6s. Another caveat of this feature was that it was able to overcome commonly faced problems in activity recognition such as occlusion, newly presented angles and changing video resolutions. However, this method does not cover scenarios in which multiple actions are being performed in the video clips.

2.1.2 Dense trajectories

The Dense Trajectories model is built on the Bag-of-Video-Feature-Words model. One of the major differences between the two methods is that the dense trajectories method uses dense trajectories method (Wang et al. 2011) to model videos as compared to the spatial-temporal features used in the Bag-of-Video-Feature-Words model. Three types of descriptors which include Histograms of Oriented Gradients (HOG) (Dalal and Triggs 2005), Histogram of Optical Flow (HOF) (Laptev et al. 2008) and Motion Boundary Histograms (MBH) (Dalal et al. 2006) are computed along these dense trajectories. The features extracted along the dense trajectories are subsequently mapped to a codebook, similar to the Bag-of-Video-Feature-Words model. The codebook is generated via the means of k-means clustering algorithm. Lastly, activity classification occurs with the help of a non-linear Support Vector Machines (SVM). The dataset used for the training of the model consists of 11 activities and a total of 1,176 video clips. The-dataset consists of both fine-grained and course-grained actions. Based on training and experimental results, the average per class accuracy for each descriptor HoG, HoF, MBH is 40%, 49%, 56% respectively, and it is 44% on average for all the descriptors combined.

2.1.3 Two-stream ConvNets

The two-stream ConvNets method utilizes Temporal Segment Networks (TSN) (Wang et al. 2016) to perform activity recognition on construction sites (Luo et al. 2018b). The method consists of four steps. The first step involves tracking the workers and drawing bounding boxes around them. MDNet (Hong et al. 2015). A single object tracking algorithm (SOT) is used to track workers in order to create temporally and spatially cropped videos. A unit of activity is deemed to be 3 seconds long. The second step focuses on extracting the spatial and temporal stream segments. Temporal streams are obtained using FlowNet 2.0 (Ilg et al. 2017) which estimates optical flow. The third step is to classify the activities with the assistance of two-stream ConvNets such as TSNs. The last step involves fusing the recognition results from spatial and temporal streams to obtain activity estimates. The dataset used for training and testing purposes consist of 16 activities in total and 1,055 clips. In terms of the average accuracy, the method outperforms the Bag-of-Video-Feature-Words and Dense Trajectories methods, resulting in an 80.5% accuracy (average of 80.8%, 79.6%, and 81.2% for three splits). The average computation time for activity classification is given in Table 1.

Table 1. Average computation time for two-stream ConvNet method

Object tracking with MDNet	2.53 frames/s
Stream creation with FlowNet 2.0	11.30s per activity unit
Activity recognition with the spatial ConvNet	0.87s per activity unit
Activity recognition with the temporal ConvNet	0.79s per activity unit

2.1.4 Three-stream CNN Model

The three-stream CNN model expands upon the idea of a two-stream approach with the addition of a third stream. The three streams are: (1) RGB stream which is used to understand spatial streams and used to identify specific background or instruments related to the activities; (2) Optical flow stream which captures temporal worker motion information from video clips; and (3) Gray streams which are used to examine edge information of moving objects and enable them to be separated into the spatial and temporal sections. The dataset used for training and testing purposes consists of 3 activities. The classification precision values for the three activities are 91%, 92%, and 100%, respectively. The average accuracy of classification using this method is about 85%.

3 METHODOLOGY

In this section, the authors talk about the activity classifier that is developed using YOLOv3 and how we designed a dataset in order to test this activity classifier. Furthermore, the authors train the methodology on our dataset and compare the test results to activity classifiers that use comparably-sized datasets. The purpose of this section is to gauge whether the proposed method can: (1) accurately identify the correct construction activity being performed; (2) provide activity classification in real-time; and (3) outperform existing activity classifiers trained on datasets pertaining to construction activities.

3.1 You only look once (YOLOv3)

You only look once (YOLO) is a state-of-the-art, real-time object detection system (Redmon and Farhadi 2018). YOLOv3 (Ver. 3) consists of two parts: (1) Darknet-53, a 53-layer network for performing feature extraction; and (2) 53 additional layers stacked onto Darknet-53 for the purpose of detection.

YOLO works by splitting up the image into a grid. Within each region of the grid, the CNN predicts the probability of the bounding box for each of the classes. One reason why the authors wanted to use YOLO as opposed to other object detection methods is because that it is very fast and efficient. As its name suggests, the algorithm only needs to look at the image once to make a prediction. Although this comes at a slight cost in accuracy and precision, that is not as critical in our context because very accurate bounding boxes around detection results are not essential. Speed is more important for our application to detect real-time images from a surveillance camera, which mostly has frame rates of around 30 frames per second (FPS), which can be handled by YOLO.

3.2 Data collection

In order to test the performance of the activity classifier and also assure that the subsequent results would help in designing the smart scheduling system, a dataset was collected and prepared. The primary criterion was to include a range of different activities that demonstrate the variety of steps that take on a construction site to complete a construction operation. Excavation was selected as a representative activity in the initial preparation stage. Screeding was selected as a representative activity in creating the base of the building. And brick laying was selected as a representative activity to construct the main exterior structure. In addition, carpentry work was selected as a representative activity that goes on around the site. All except excavating are human activities. To create the dataset, the authors first searched for videos on YouTube that included each of our activities. The summary of videos accumulated for the purpose of training and testing the classifier are shown in Table 2.

Table 2. Summary of proposed dataset

Activity	Numberof clips	Mean clip length (s)	Standard deviation(s)
Excavating	20	10.8	5.5
Screeding	20	9.5	3.4
Bricklaying	20	8.9	3.7
Carpentry	20	8.1	3.5

The proposed data on average has 20 videos for each activity, with a mean clip length of 9.3s and standard deviation of 4.1s. One of other criteria was to have a dataset that was comparable to datasets used in previously proposed activity classifiers trained specifically to perform activity recognition on construction sites. Table 3 shows a comparison of our dataset versus the ones used in other activity classifiers for construction activities.

Table 3. Comparison of proposed dataset and existing datasets

Method	Number of activities	Number of clips	Average clips per activity	Average length of clip (s)
Gong et al. (2011) (Dataset II)	5	300	60	5
Yang et al. (2016)	11	1176	107	6.8
Luo et al. (2018a)	3	654	218	8.8
Luo et al. (2018b)	16	1055	66	3
Ours	4	80	20	9.3

3.3 Dataset

The dataset we built consists of 4 classes: Excavating, Concrete Screeding, Brick Laying, and Carpentry (Figure 1). Once the videos are collected, every 3rd frame from each of these videos are extracted. Because for most of our videos, there was only a significant change in movement after every 3rd frame because of the frame rate and type of activity. Getting all the frames of a 7 second video would give us around 7×30 = 210 frames. There would be too many redundant frames and this would lead to unnecessary extra cost to annotate and label. By taking every 3rd frame, we get around 7 × 30 / 3 = 70 frames for each video. The authors then labeled the class and annotated the bounding boxes for each of these frames. Once the annotated frames for each activity are developed, we used an 80-20 split to get the training and testing sets. For each activity, 80% of the frames are used for training, and the remaining 20% are used for testing in YOLO. Lastly, a TXT file with the paths to each of images was created, so that YOLO learns where to look for each image, and its corresponding annotations.

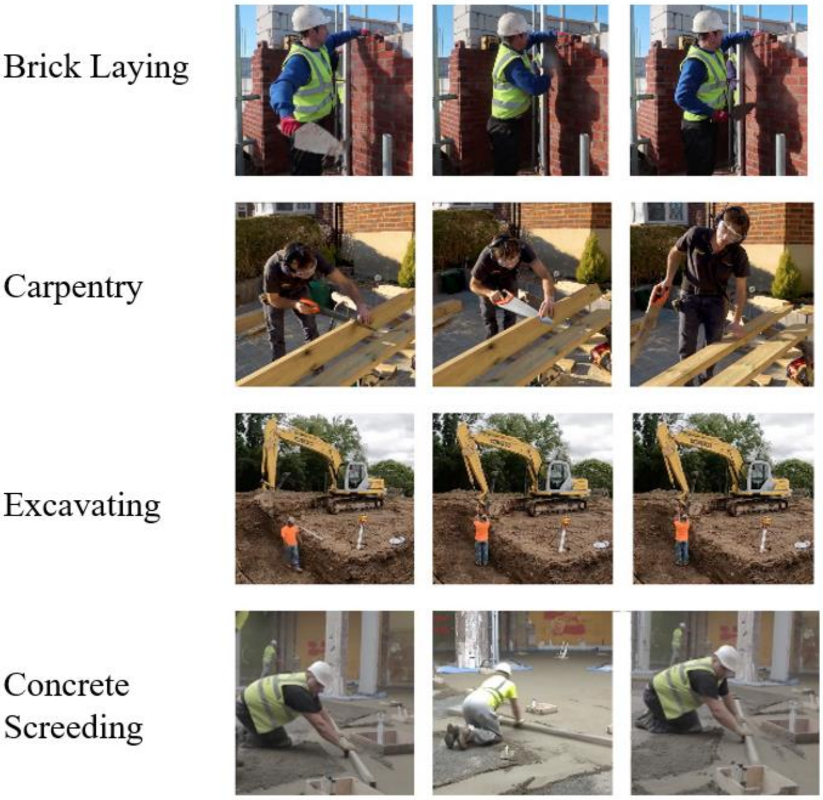


Figure 1. Example activities in prepared dataset

3.4 Developing our YOLOv3 classifier

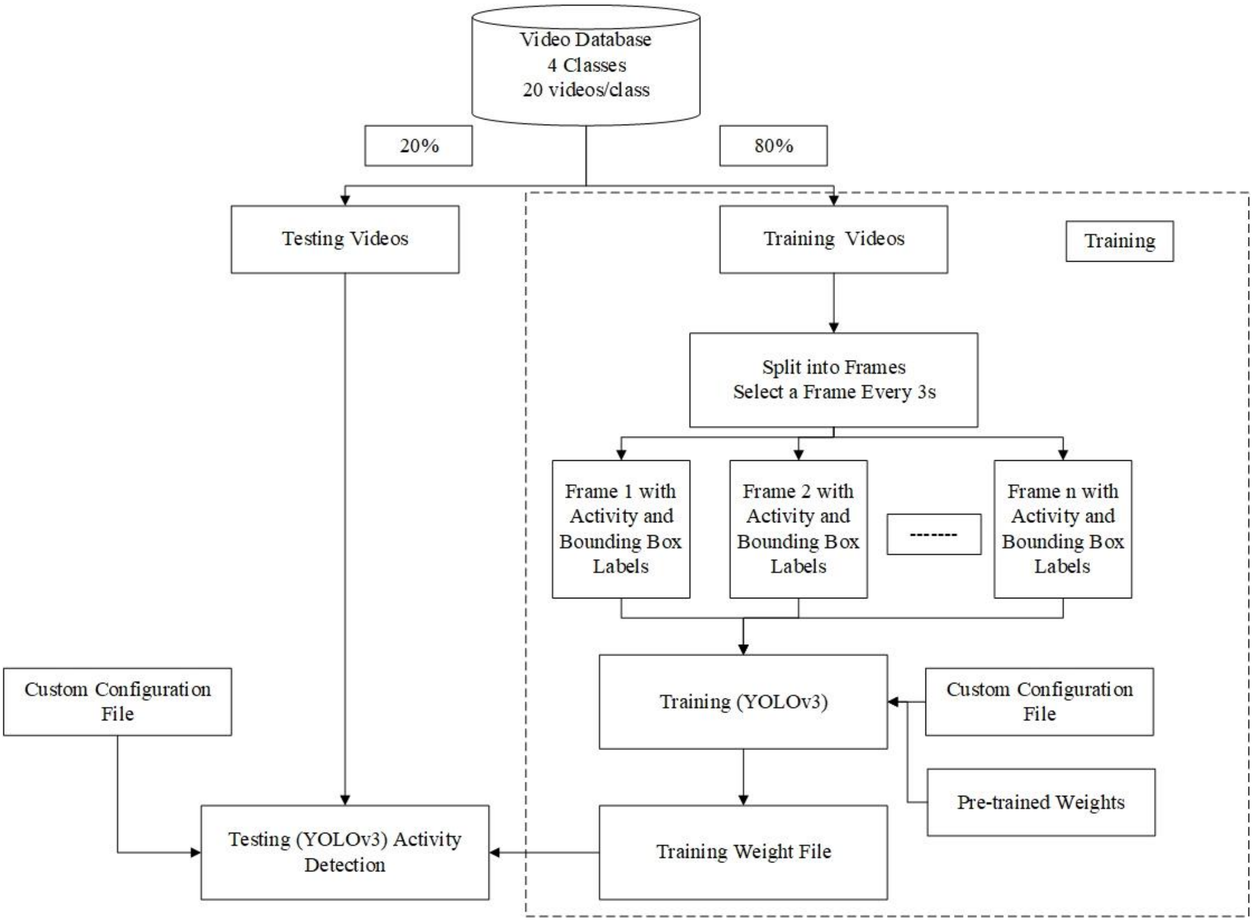
The first step in developing the classifier involved the splitting of the dataset into training and testing sets for which we used an 80-20 split. Each training video was passed through a script that splits the video into frames. For the purpose of the dataset preparation, the authors selected every 3rd frame and discarded the other two. Each frame was individually labelled and a TXT file of the same name as the frame was created for each corresponding frame, including the class and the bounding box coordinates. In order to perform training on the dataset, few changes needed to be made to the YOLOv3 configuration file (yolov3.cfg) and the text files that are used for training, including:

- Number of filters = (Number of classes + 5) * 3
- train.txt, a TXT file containing the file paths of the training frames
- test.txt, a TXT file containing the file paths of the testing frames
- obj.names, a file containing a list of all the classes
- obj.data, a file containing number of classes and the file paths of train.txt and test.txt files

Along with the modified configurations, the following information is also required:

- Pre-trained convolutional weights
- A configuration file containing the layers present in YOLOv3

A flowchart of the training procedure is given in Figure 2


Video Database
4 Classes
20 videos/class
20%
80%
Testing Videos
Training Videos
Training
Split into Frames
Select a Frame Every 3s
Frame 1 with Activity and Bounding Box Labels
Frame 2 with Activity and Bounding Box Labels

Frame n with Activity and Bounding Box Labels
Custom Configuration File
Training (YOLOv3)
Custom Configuration File
Pre-trained Weights
Testing (YOLOv3) Activity Detection
Training Weight File

Figure 2. Processing of training process to obtain weights file

3.5 YOLO results and comparison

The training and object detection algorithms were run on a Google Colab Notebook with the following specifications:

- OS: Ubuntu 18.04.3 LTS (64 bit)
- CPU: Intel(R) Xeon(R) @ 2.20GHz
- CPU RAM: 12GB
- GPU: Nvidia Tesla K80
- GPU RAM: 12GB GDDR5

The authors ran training for up to 11000 iterations with each iteration taking 13-18 seconds. The results from the iterations are shown in Table 4. Using the testing set, the detection time was 63.0 seconds for 564 images in the testing set. It demonstrates a detection rate of 29/564 = 51ms per image. This results in a frame rate of 19FPS. This would be usable with surveillance cameras that usually operate at 30FPS. As discussed before, the authors don't need to view every frame of the video feed. Taking into account the additional computational overhead of our smart scheduler, this would be feasible for a construction site application. Figure 3 demonstrates a successful detection for carpentry activity (9,440 iterations).

Table 4. Results for different iterations based on Equation (1) – (3)

Iterations	mAP	Precision	Recall	F1 Score
9440	60.28	82	49	61
8580	43.58	97	27	42
9720	43.14	69	43	53
9860	57.21	66	48	56
10000	55.50	75	49	59
10140	45.48	71	38	48
10280	47.28	66	38	48
10420	43.89	73	40	52
10560	48.87	72	45	55
10700	57.64	60	50	54
10840	33.83	35	40	38
10980	44.31	72	38	50

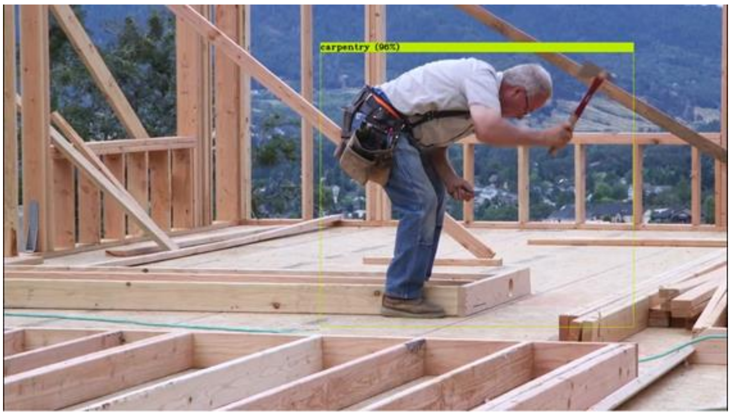


Figure 3. Successful detection of carpentry activity (9440 iterations)

In Table 4, the authors showed results for the detection on the dataset with various metrics. However, the authors want to identify which metric is most meaningful for our application.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad (2)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad (3)$$

For instance, having higher number of false positives (FP) would decrease the precision. A high number of FPs means we are detecting objects as a certain activity, when in fact that activity is not taking place in those objects. To a construction manager, this would look like more work is being done on the site than what is actually taking place. The result is the work would seem ahead-of-schedule whereas it may not. Succeeding tasks may therefore be scheduled before they should be and this would cause inefficiencies.

On the other hand, having a high number of false negatives (FNs) would decrease the recall. A FN happens when we don’t detect an activity that is actually taking place. Certain tasks being done on the site may not be detected as they should be. And this would therefore make it seem like the work is behind schedule, when in fact, the work may be on track.

The balance between Precision and Recall is given by the F1 Score. In this research, the authors want to minimize FPs; the cost of misinformation from a FP is greater in comparison to a FN. Minimizing the FPs means looking for a higher precision value, while still making sure that the F1 Score is acceptable. The best result that is obtained on our dataset as seen in table 4 was at 9440 iterations. At this level, the authors have the highest mAP and F1 Score, while also having the second highest precision. Additionally, this level also has the second highest recall. For this iteration level, Table 5 shows the results for each class. Then our results are compared to that of other state-of-the-art models (Table 6). These are the same papers as those shown in Table 3.

Table 5. Results for each class at 9440 iterations

Class	Precision (%)
Excavating	76.56
Cement Screeding	93.75
Brick Laying	81.54
Carpentry	61.97

Table 6. Comparison to other models

Method	Average Accuracy (%)
Gong et al. (2011)	79
Yang at al. (2016)	59
Luo et al. (2018a)	85
Luo et al. (2018b)	81
Ours	78

4 SMART SCHEDULE MONITORING

In this section, the authors demonstrate how we use the proposed activity classifier to perform smart schedule monitoring of construction projects. The purpose of the section is to: (1) demonstrate the concept of smart schedule monitoring; (2) explain how the proposed activity classification method can aid in the process of smart schedule monitoring; and (3) gauge whether this method of activity monitoring is feasible to implement on construction sites.

4.1 Distribution into project phases

The construction activities for the activity classifier were chosen keeping in mind that a construction project involves different phases and activities following certain precedence relationships. The four activities chosen for the dataset were Excavating, Carpentry, Bricklaying and Cement screeding, respectively. For the purpose of the schedule monitoring system, the authors distributed the activities into three phases as shown in Table 7.

Table 7. Distribution of activities into construction phases

Phase 1	**Phase 2**	**Phase 3**
Excavating	Carpentry Bricklaying	Screeding

Once the phases of the construction project were defined, the next step was to represent this information in the form of a Gantt Chart (Maylor 2001). As shown in Figure 4, the construction project is divided into 3 phases. Each task has a certain time period assigned to it. The number below each period of task completion represents the number of workers assigned to the task. This information is provided by the project manager at the beginning of the construction project.

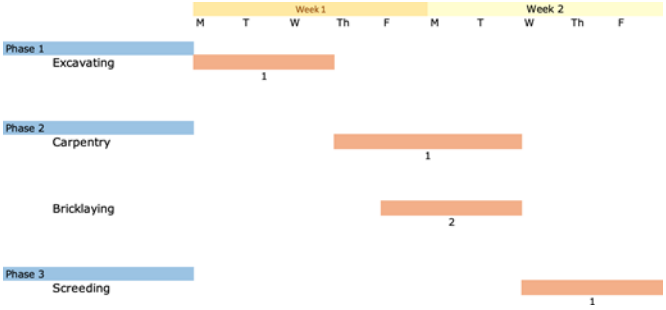


Figure 4. Example of Gantt Chart showing different phases of project

4.2 Merging the two concepts

Now that: (1) the tasks are represented in terms of a timeline; and (2) the tasks being performed on the site can be identified using the proposed activity classifier, the next step in the process is to combine the two concepts. Firstly, the video feed from the camera placed on the construction site is passed to the activity classifier. The activity classifier then determines what activity is being performed in the clip and how many workers are performing the task. Once we know the task being performed, this information along with the original project timeline is fed into the smart schedule monitoring system to update the start and finish times of the project automatically and accordingly adjust the schedule of the project. A flowchart of this process is carried out as shown in Figure 5.

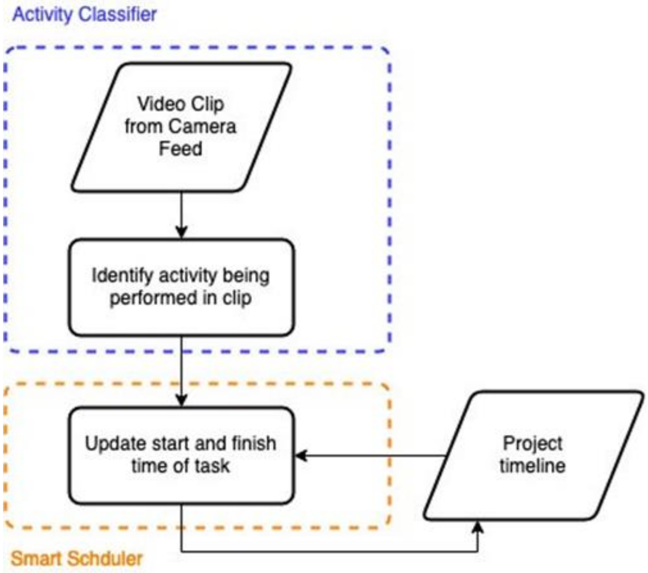

Activity Classifier
Video Clip from Camera Feed
Identify activity being performed in clip
Update start and finish time of task
Project timeline
Smart Schduler

Figure 5. Flowchart of Smart Schedule Monitoring

Figure 6 shows an example in which start and finish times are updated based on what activity is detected on the video camera feed. The current state in the timeline is represented by the dashed line. In this scenario Phase I (Excavating) and Phase II (Carpentry) were detected to be on time and updated on the Gantt Chart, represented by the green markings below the activity bars. However, on the current time stamp, the activity detected is bricklaying (Phase II), not screeding (Phase III) as expected. Based on this detection, the timeline is now updated to reflect a new finish time for bricklaying as shown by the extended task line for the bricklaying activity. The scheduler also updates the start time for the task of screeding. The number of workers detected is also one, and this is reflected on the project timeline thus helping the project manager to reallocate resources accordingly.

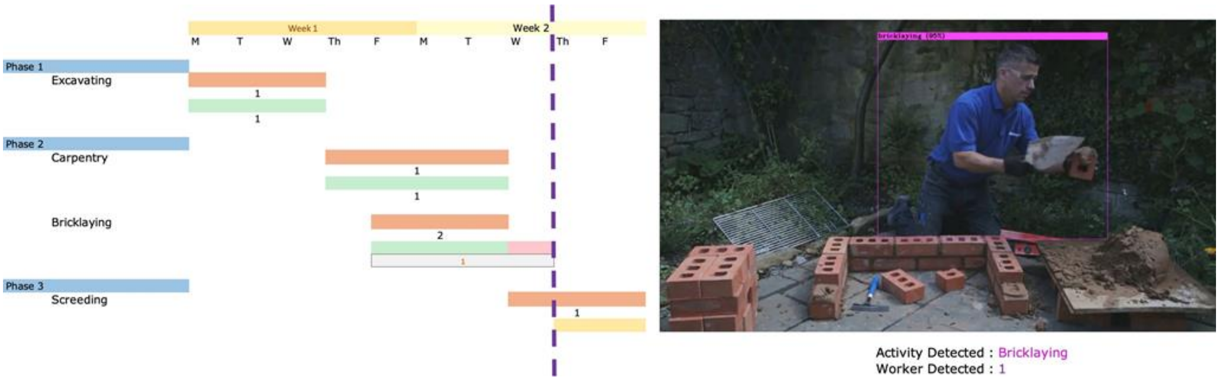


Figure 6. Example of updating start and finish times

5 DISCUSSION

The application that was developed in this paper is very versatile in that many additional features and improvements can be added to it. With a basic setup and understanding of the current model, there are many changes that can be made to further increase the potential of such an application. A goal of many software development is to be configurable to each user's needs. Currently, our implementation is very fixed in terms of the classes that it can detect and ways in which it can be used.

5.1 Activity classifier

As shown in Table 3, our dataset is still smaller than most other datasets. Although our average clip length is longer in comparison to others', it has fewer number of clips. This means that there isn't much variety in the video clips and that could reduce the amount of learning that YOLO can do from these images. The longer length of our clips doesn't necessarily help, because the activity taking place is just being repeated more often, resulting in very similar frames. Having more video clips, and hence, more frames would help improve the results.

Building our dataset based on YouTube videos proved challenging itself. It was even harder to find videos in which multiple activities were taking place. Therefore, our dataset didn't contain images where multiple activities are labeled in the same frame. This limits the learning that YOLO can do. The situation where more than one activity occurs has not been tested due to the lack of available images of that sort.

5.2 Smart scheduler

The smart Scheduler is fully usable on it's own. However, most major construction projects rely on a centralized Building Information Modelling (BIM) systems. These are 4D visualization applications that are used to show the progress of a construction site in 3D over time.

It allows project managers to understand how progress is expected to occur on the site and is referred to throughout the construction project. Being able to integrate the scheduler with it would be a very useful goal. It would allow automatic updates to the BIM system's progress tracking. It would also allow real-time tracking of the progress which previously has been a challenge. A manager would no longer need to manually update progress in order to see progress as it would be updated automatically in real-time.

Additionally, a further level of complexity can be added to the scheduler which allows to track what is being done by each worker specifically. This allows them to be reassigned based on their performance and skills in certain activities. For instance, if a worker is able to complete a specific task faster and ahead of the planned schedule, they would be considered efficient at that task. The scheduler can then make recommendations on which personnel should work on which tasks.

6 LIMITATIONS

As discussed above, the authors ran training for up to 11000 iterations on the dataset. As shown in Table 4, the result of more training was inconclusive in terms of increasing accuracy. Training for much longer, over 25000 iterations may give us a better view.

Additionally, our dataset consists of un-occluded images for training and testing and so we haven’t done testing with occluded images to see how our model reacts in that scenario. However, even if the authors had occluded images, YOLO itself isn’t very good out-of-the-box to deal with occlusion.

The smart schedule monitoring is currently also limited to detecting classes that are predefined. These classes don’t automatically match the activities being trained by YOLO. YOLO takes in a file with a list of all the classes. This file could also be used to update the list of available activities for the schedule monitoring, in which the manager can choose to use.

7 CONCLUSION

As seen in our results section, our dataset, although with its issues, has been successful in helping us train and build a model that can be used for activity classification on a construction site. The authors have further enhanced the usage of these detections by using them to predict, track and update the progress that is being made on the construction site. By allowing project managers to see real-time changes to the progress being made, they can much more quickly and easily find ways to reallocate labor and other resources to keep projects on track.

By creating such a system where data is collected real-time, we can help all stakeholders of the project monitor the progress more easily. This includes the workers themselves. They can see if at their current productivity, they will be able to complete the task on time. This allows workers to be more mindful and aware of their progress. It can also help alleviate them from having to be told to speed up their work as they have access to this information themselves.

Most importantly, this solution will help identify delays as they start to develop. As soon as a delay is detected in the smart schedule monitoring system, managers can be notified and they will be able to see what the problem is, and then decide on how they can fix it. This will help prevent inefficiencies on the site as soon as they occur and let them be addressed immediately. That is where real-time tracking is useful and value adding.

The implication of a reduction in construction delays can be very significant. It would lower costs for the firm such as by reducing labour hours needed and decreasing the tool and equipment leasing expenses. These all can amount to a large part of construction costs (Gouett et al. 2011). This would be an important investment for firms to help them reduce these costs. There would also be positive economic impact and environmental impact. According to the World Green Building Council (WGBC), the construction processes account for about 11% of global carbon emissions (Council 2019). Reducing the time spent in construction means a reduction in transportation of materials and less waste as well. All of these factors could lead to less pollution and a smaller carbon footprint.

Most construction sites already have surveillance cameras for their security purpose, and integrating this to work with multiple camera video feeds would be important in building a reliable and functional system. Additionally, the data collected from each construction project can be used by the construction company for training and analysis, so that they can improve their workflow in future projects. This data-centric approach is a shift towards the right direction for advancing the technologies used on a construction site.

Integrating our methodology with a BIM software tool would be a very good way for many of these goals to be achieved. It will give managers a much more centralized view of all their project statistics and will automatically make adjustments to the plan based on the real-time progress. By using software like this, project managers will be able to identify causes of delays. Automatic updates of the schedule based on the real-time tracking of activities will aid in management of resources. It will help them in planning and logistics of the project, while reducing costs.

REFERENCE

Abbasi, O., Noorzai, E., Gharouni Jafari, K., and Golabchi, M. (2020). Exploring the causes of delays in construction industry using a cause-and-effect diagram: case study for Iran. *Journal of Architectural Engineering*, 26(3), 05020008.

Agyekum-Mensah, G., and Knight, A. D. (2017). The professionals' perspective on the causes of project delay in the construction industry. *Engineering, Construction and Architectural Management*.

Akhavian, R. and Behzadan, A. (2015a). Wearable sensor-based activity recognition for data-driven simulation of construction workers' activities. *In 2015 Winter Simulation Conference (WSC) (IEEE)*, 3333–3344

Akhavian, R. and Behzadan, A. H. (2015b). Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers. *Advanced Engineering Informatics* 29, 867–877

Akhavian, R. and Behzadan, A. H. (2016). Smartphone-based construction workers' activity recognition and classification. *Automation in Construction* 71, 198–209

Akhavian, R. and Behzadan, A. H. (2018). Coupling human activity recognition and wearable sensors for data-driven construction simulation. *ITcon* 23, 1–15

Akhavian, R., Brito, L., and Behzadan, A. (2015). Integrated mobile sensor-based activity recognition of construction equipment and human crews. *In Proceedings of the 2015 Conference on Autonomous and Robotic Construction of Infrastructure*. 1–20

Alwi, S., and Hampson, K. D. (2003). Identifying the important causes of delays in building construction projects. In *The 9th East Asia-Pacific Conference on Structural Engineering and Construction*.

Blanco, J. L., Mullin, A., Pandya, K., and Sridhar, M. (2017). The new age of engineering and construction technology. *McKinsey & Company-Capital Projects & Infrastructure*.

Cheng, C. F., Rashidi, A., Davenport, M. A., and Anderson, D. (2016). Audio signal processing for activity recognition of construction heavy equipment. *In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC Publications)*, vol. 33, 1

Council, W. G. B. (2019). New report: the building and construction sector can reach net zero carbon emissions by 2050

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (IEEE)*, vol. 1, 886–893

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. *In European conference on computer vision (Springer)*, 428–441

Durdyev, S., Omarov, M., and Ismail, S. (2017). Causes of delay in residential construction projects in Cambodia. *Cogent Engineering*, 4(1), 1291117.

Gong, J., Caldas, C. H., and Gordon, C. (2011). Learning and classifying actions of construction workers and equipment using bag-of-video-feature-words and bayesian network models. *Advanced Engineering Informatics* 25, 771 – 782. doi:https://doi.org/10.1016/j.aei.2011.06.002. Special Section: Advances and Challenges in Computing in Civil and Building Engineering

Gouett, M. C., Haas, C. T., Goodrum, P. M., and Caldas, C. H. (2011). Activity analysis for direct-work rate improvement in construction. *Journal of Construction Engineering and Management* 137, 1117–1124

Hong, S., You, T., Kwak, S., and Han, B. (2015). Online tracking by learning discriminative saliency map with convolutional neural network. *In International conference on machine learning*. 597–606

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition.* 2462–2470

Joshua, L. and Varghese, K. (2011). Accelerometer-based activity recognition in construction. *Journal of computing in civil engineering* 25, 370–379

Kao, T.-P., Lin, C.-W., and Wang, J.-S. (2009). Development of a portable activity detector for daily activity recognition. *In 2009 IEEE International Symposium on Industrial Electronics (IEEE)*, 115–120

Laptev, I. (2005). On space-time interest points. *International journal of computer vision* 64, 107–123

Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. *In 2008 IEEE Conference on Computer Vision and Pattern Recognition (IEEE)*, 1–8

Luo, H., Xiong, C., Fang, W., Love, P. E., Zhang, B., and Ouyang, X. (2018a). Convolutional neural networks: Computer vision-based workforce activity assessment in construction. *Automation in Construction* 94, 282 – 289. doi:https://doi.org/10.1016/j.autcon.2018.06.007

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

Maylor, H. (2001). Beyond the gantt chart:: Project management moving on. *European management journal* 19, 92–100

Mpofu, B., Ochieng, E. G., Moobela, C., and Pretorius, A. (2017). Profiling causative factors leading to construction project delays in the United Arab Emirates. *Engineering, Construction and Architectural Management*.

Mishra, A. and Bhandari, S. (2018). Performance assessment of ongoing construction projects under town development fund, *nepal* 1, 27–39

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint*

arXiv:1804.02767

Wang, H., Kla¨ser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. *In CVPR 2011 (IEEE)*, 3169–3176

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2016). Temporal segment networks: Towards good practices for deep action recognition. *In European conference on computer vision (Springer)*, 20–36

Yang, J., Shi, Z., and Wu, Z. (2016). Vision-based action recognition of construction workers using dense trajectories. *Advanced Engineering Informatics* 30, 327 – 336. doi:https://doi.org/10.1016/j.aei.2016.04.009