2021 European Conference on Computing in Construction
Online eConference
July 26-28, 2021

JOINT DETECTION AND ACTIVITY RECOGNITION OF CONSTRUCTION WORKERS USING CONVOLUTIONAL NEURAL NETWORKS

Ghazaleh Torabi, Amin Hammad, Nizar Bouguila
Concordia University, Montreal, Canada

ABSTRACT

Manually gathering information about activities on construction sites for project management purposes is labor-intensive and time-consuming. As a result, several works leveraged the already installed surveillance cameras to automate this process. However, the recent learning-based methods discretize continuous activities by assigning a single label to multiple consecutive frames. They do not fully leverage the contextual cues in the scene, and are not optimized end-to-end. A variation of the YOWO network, called YOWO53, is proposed in this paper to address these limitations. YOWO53 shows better classification and detection results over YOWO and allows using smaller input frames with real-time speed.

INTRODUCTION

Having access to detailed information about construction sites is beneficial to many project management tasks including productivity analysis. Unfortunately, manually gathering this information is labor-intensive, time-consuming, and may not be detailed or efficient enough, especially for large projects. Luckily, surveillance cameras are already installed in most construction sites nowadays. As a result, several works leveraged them to address this issue by using vision-based automatic activity recognition methods (Luo et al., 2020, 2019, 2018a, 2018b). Automation helps to analyze potential reasons for idling and productivity loss; but the existing methods are still far from being applicable to the real world.

The recent learning-based frameworks such as the ones used in (Luo et al., 2020, 2019, 2018b) consist of three main modules, detection, tracking, and activity recognition; each trained separately. These frameworks are referred to as the three-stage frameworks throughout this paper. They start with a worker detection module detecting every worker in every frame, followed by a tracking module that connects the detected workers in consecutive frames. This results in cropped video clips that contain a single worker performing a certain task (Figure 1). The video clips are then automatically broken into short segments (16 or 64 frames) by the activity recognition module, and the type of activity performed by single workers are recognized for each segment.

One of the shortcomings of these frameworks is that the activity recognition modules discretize continuous activities, and produce a single label for each discrete segment. In other words, these frameworks perform segment-level (e.g. 16, or 64-frame segments) activity recognition; assuming that each discretized segment contains a single activity. These activity recognition modules are trained and tested on video clips that are manually trimmed around single activities. Therefore, they fail when this ideal situation is not satisfied in the real-world due to unsupervised automatic discretization of site videos. Despite using short segments to make sure that they span no more than one activity, there is no guarantee that all the frames of each segment are placed in the middle of an activity and do not contain, for example, the ending of one as well as the beginning of another activity. In addition, the modules are optimized separately, which does not guarantee the optimization of the entire framework. For example, the detected boxes around workers may be too tight resulting in video clips containing no context or background information while some activities are easier to detect having that additional information (e.g. activities containing interaction with a tool/object or another worker, or activities that only take place in a specific part of the construction site). Note that the performance of the three-stage frameworks is lower than the performance of the individual modules as the error propagates from the first to the last module; a wrong or incomplete detection will result in a wrong activity recognition.

Context is another matter that has been missing so far in the previous construction-related activity recognition works. There have been several papers in computer vision such as (Pan et al., 2020), investigating the improvement brought to activity recognition by using context; but they are rarely leveraged by the construction community. One of the few works using contextual information to improve the activity recognition module on construction sites is (Luo et al., 2020). They used additional modules on top of the three-stage framework to leverage the information from nearby workers assuming workers are working in groups. Still, the entire framework is not optimized end to end, other contextual information is not considered, and its performance is conditioned on the number of nearby workers, as well as the assumption that the activity of nearby workers are somehow related. Having said that, the improvement brought by the additional module supports the claim that contextual information is beneficial to activity recognition.

Latency is another issue with the three-stage frameworks. In these frameworks, the output of each module needs to be saved somewhere in the memory, processed, and then be fed to the next module, leaving the next module idling during this process. The latency becomes more significant when we are dealing with about 7-10 hours of site videos per day.

This study aims to apply a fast, and fully optimized CNN-based method (You Only Watch Once, YOWO (Köpüklü et al., 2020)) to jointly detect construction workers and their activities in every frame; and then track the results through the entire video. A variation of this network with more accurate detections is proposed in this paper to address the challenge of detecting small workers in large construction site video frames. The two networks have been tested with various 3D backbones. The full frames are used for both the detection and activity recognition, allowing the networks to extract useful contextual information from the scene and minimizing the effect of wrong detections on activity recognition. The networks recognize continuous activities in each frame (frame-level activity recognition), instead of discretizing the activities. Using a single module enables the optimization of the entire framework end-to-end. In addition, a sensitivity analysis is applied to compare the results of the two networks with different 3D backbones and input frame sizes with respect to accuracy, precision, recall, and speed.

LITERATURE REVIEW

Video understanding methods can be classified based on their input or output types. Input videos can be trimmed, or untrimmed while the output of the methods depends on the task at hand.

Trimmed videos are temporally trimmed around single activities and the goal is to produce a single label for the entire clip. Untrimmed videos can contain some unrelated frames, or even multiple activities. They are easier to gather with no manual effort allowing the creation of large datasets. However, activity recognition for these videos is generally more challenging and shows lower performance (Heilbron et al., 2015). There has been extensive research on both input types in computer vision; but the construction community has been mainly focused on using single activity trimmed inputs to train the models, which is not a practical assumption when it comes to real-world problems as explained in the previous section.

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, whether it is trimmed or untrimmed (Heilbron et al., 2015). This method is the ideal solution when the input video contains a single actor performing a fixed activity in the entire video. The construction community has been using this approach to recognize discretized activities in construction site videos so far (Luo et al., 2020, 2019, 2018a, 2018b). 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.

Spatiotemporal activity detection is another video understanding task; and is used when multiple actors are performing different tasks in trimmed or untrimmed videos. Spatiotemporal activity detection methods produce bounding boxes around workers as well as their activity labels (Girdhar et al., 2019; Kalogeiton et al., 2017; Köpüklü et al., 2020; Pan et al., 2020; Yang et al., 2019). Considering the nature of construction site videos, spatiotemporal activity detection is the most suitable method.

There have been few studies on the activity recognition of workers on construction sites. Authors of (Luo et al., 2018a) used Faster R-CNN to detect workers and construction-related objects in still site images. They construct 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 are inferred. Since the decision is made based on still site images, temporal information is not taken into account. They later Introduced a framework in (Luo et al., 2018b) to produce activity labels for all workers in 3-second video clips by considering the temporal information. They specify workers bounding boxes manually and track them with a single object tracking algorithm. Next, 3-second video clips are divided into three segments. One RGB frame is chosen randomly from each segment and fed to Temporal Segment Network (TSN) (Wang et al., 2017) along with five random consecutive optical flow frames. TSN predicts the probabilities of each class, and the activity with the highest score is chosen among 16 classes for the entire 3-second video clip. Furthermore, to evaluate the productivity of the workers, they classify the workers' activities into three classes of productive, non-productive, and semi-productive.

Aligning worker groups with workspaces in advance helps with improving the performance and safety of workers. This requires detection and understanding of dynamic workspaces. To visualize workspaces, authors in (Luo et al., 2019) first detect and track workers through 3-second video clips. YOLOV3 (You Only Look Once) (Redmon and Farhadi, n.d.) together with a multiple object tracking method was used for this purpose. The clips were spatially cropped around the smallest bounding boxes that surround detected workers regardless of their movement in the clips. ResNext-101 (Hara et al., 2018) was then fed with extended bounding boxes separately to recognize 12 different actions. The activity locations were then projected from the frame plane into floor plan coordinates. Assuming that action classes and their

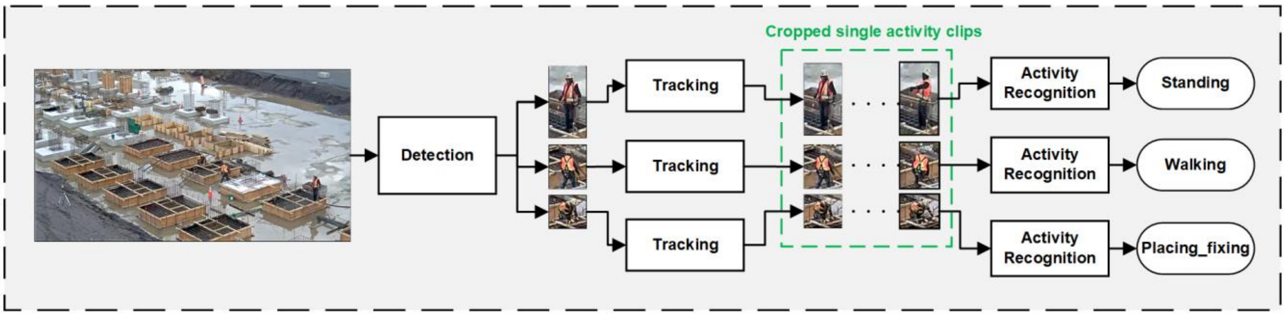


Figure 1: Detection-Tracking-Activity recognition workflow

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 (Ankerst et al., 1999)) was used to group area points together.

To detect the activities of workers in groups, the authors of (Luo et al., 2020) combined deep features and contextual information. They used YOLOV3 for workers detection, SORT (Bewley et al., 2016) multiple object tracking method, and ResNext-101 to extract deep features from single worker clips. They defined the spatial distance between workers based on the overlap and distance of workers bounding boxes and then applied k-nearest neighbors to generate an activity graph that shows the relevance of workers to each other. Finally, they input the activity graph and deep features to a conditional random field (CRF) and inferred the most probable activity among 17 classes for each worker, based on their relationship with the neighboring workers as well as the deep features extracted with ResNext-101.

Figure 1 shows the main structure of the activity recognition framework used in (Luo et al., 2020, 2019, 2018b) excluding the additional CRF post-processing in (Luo et al., 2020). As mentioned at the beginning of this section, all of these methods consider the input to be spatially cropped around single workers, and discretize the continuous activities. Assuming that the discretized segments are temporally trimmed around single activities, they produce a single label for the entire duration of segments (segment-level activity classification). Regardless of the short duration of segments, this assumption does not always hold when applying the method to real-world problems. To apply these methods to real construction site videos, the videos are automatically segmented by the activity recognition module. However, there is no manual supervision to make sure these segments contain single activities. In addition, all of the above papers (expect (Luo et al., 2018a), which does not consider temporal information at all), use separate detection and activity recognition modules. However, separate optimization of these modules does not guarantee the end-to-end optimization of the framework. Moreover, contextual information is only considered in (Luo et al., 2020) for workers working in groups. The context used in (Luo et al., 2020), is only the activity of nearby workers; and other objects, tools, or the general scene are not used to improve the performance.

METHOD

Dataset preparation and annotation

Surveillance cameras can be found in most construction sites nowadays. These cameras are typically used to deter unwanted, illegal, or dangerous activities, as well as to protect against robbery. By recording and storing site videos, hours of data can be generated and used in computer vision applications.

There are multiple trades on construction sites including carpenters, bricklayers, concrete workers, metal workers, etc. Depending on the type of the construction project, a subset of these trades and activities may exist in the site videos. The project schedule coupled with the Building Information Model (BIM) can be used to identify the type of the project, existing trades, and activities at different phases of the project. Site videos can later be studied to confirm the selected activities and pick the ones that are visible to each installed camera to train the network. On the other hand, there are generic activities, such as walking, standing, sitting, and transporting tools and materials, that are done regardless of the type of the project. These activities are present in almost all construction sites and can be generalized from one site to another. For example, the construction project used in this research was mainly formwork, concrete, and steelwork, but included the above generic activities as well.

The training input videos to the method used in this paper, are short video clips containing multiple workers performing different continuous activities; as opposed to the cropped single activity clips shown in Figure 1. To generate the dataset, video clips containing the activities of interest are trimmed from the site videos. There is no need to ensure that the training clips contain single discrete activities or single actors; but the trimming is still done so that only the portions of the site videos containing activities of interest are included in the dataset (consecutive activities can exist in the training clips as long as they are annotated). Next, clip frames are extracted and annotated with bounding boxes around workers as well as their activities.

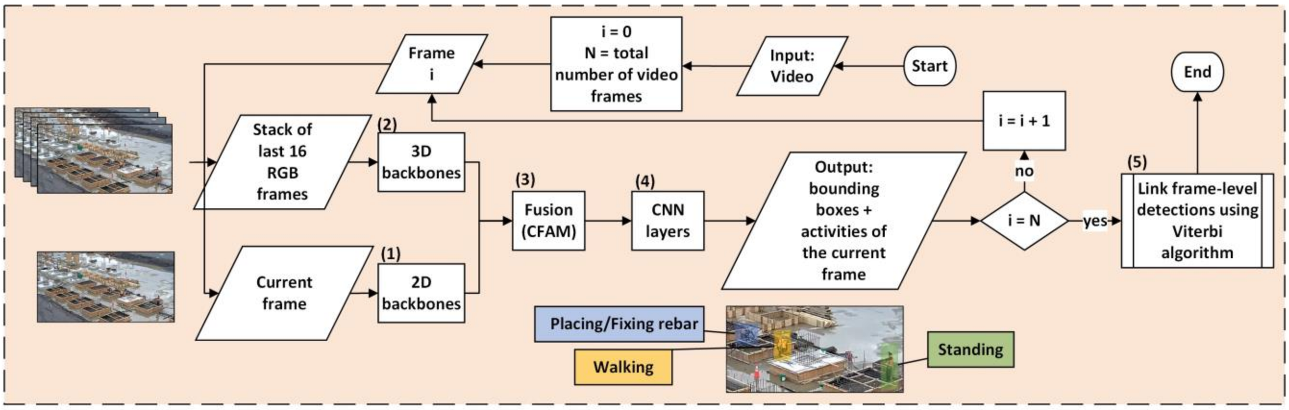

Frame i
i = 0
N = total number of video frames
Input: Video
Start
End
Stack of last 16 RGB frames
(2)
3D backbones
i = i + 1
no
(5)
(3)
(4)
Fusion (CFAM)
CNN layers
Output: bounding boxes + activities of the current frame
i = N
yes
Link frame-level detections using Viterbi algorithm
(1)
Current frame
2D backbones
Placing/Fixing rebar
Walking
Standing

Figure 2: The overall activity recognition framework of YOWO53

Detection and activity recognition

In contrast to the methods introduced in the literature review, the method used in this paper uses the entire frame as input (instead of bounding boxes) to recognize activity classes. Hence, more contextual information is considered, and recognitions are more robust to detection errors (i.e. a wrongly placed bounding box may not contain the entire body of workers, hence the resulting clips are not suitable to be used for activity recognition and will propagate the error). The detection and recognition are done jointly and are optimized end-to-end. Therefore, the final error includes the error from both stages, giving a better understanding of the performance of the entire framework.

The network used in this paper is a slight variation of YOWO (Köpüklü et al., 2020) called YOWO53. The original YOWO is a spatiotemporal activity recognition network that jointly detects and recognizes continuous activities for each frame (frame-level activity recognition) instead of discretizing them into single activity segments (segment-level activity recognition).

As shown in Figure 2, YOWO53 consists of five main blocks: (1) 2D backbone, (2) 3D backbone, (3) Channel fusion and attention mechanism (CFAM), (4) Output CNN layers, and (5) Linking. Each block is briefly explained below.

2D backbone: The 2D backbone extracts spatial information from the current frame using 2D CNNs. The detection results are mainly affected by this branch. The original YOWO introduced in (Köpüklü et al., 2020) uses the YOLOV2 (Redmon and Farhadi, 2016) backbone network (Darknet19) for this block. Darknet19 is fast, but not very accurate in detecting small objects. YOLOV3 (Redmon and Farhadi, n.d.) backbone (Darknet53), is an improved version of Darknet19 with more accurate detections for small objects and a slightly lower speed. Since cameras are usually installed at a height to cover large portions of the construction sites, workers appear very small in the site videos. Therefore, the new version of YOWO introduced in this paper uses Darknet53 instead, to improve the detections.

3D backbone: The 3D backbone uses the last 16 frames (including the current frame) to extract temporal information which mainly helps with activity recognition. Different networks have been tested for this block in (Köpüklü et al., 2020). ResNext-101, due to its high accuracy, and ShuffleNetV2 2x (Köpüklü et al., 2019) due to its high speed, are chosen among them for this paper.

CFAM: The output shape of both 2D and 3D branches are the same and they are concatenated and fed to the CFAM block for fusion. CFAM uses the Gram matrix-based attention. First, the concatenated feature maps are processed through convolutional layers. Then, they are reshaped and multiplied with their transpose to find the correlation between different channels. Finally, they go through a softmax layer resulting in attention weights that are multiplied with, and added, to the original feature maps. Using this method, each channel contains the summation of its own initial features as well as the weighted features of the rest of the channels.

Output CNN layers: The fused feature maps are processed through additional convolutional layers followed by a regression module. Finally, a 1x1 convolutional kernel with N channels is applied to it to find the probability of each activity class, center point (x,y) offsets, height offset, and width offset of each anchor box, as well as the detection confidence score. N is given in (1), and the number of anchor boxes is set to five in YOWO53 following the same set up of YOWO.

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

Linking: Detected boxes from consecutive frames are finally joined based on their class scores and IoU (intersection over union) using the Viterbi algorithm.

Construction sites are large compared to the size of workers. Moreover, cameras are usually placed at a height to have a large field of view. Therefore, workers appear very small in the frames, and using high-resolution frames is essential for worker detection, activity recognition, and even annotation. Increasing the size of the frame will reduce the speed considerably. Therefore, this paper applies a sensitivity analysis to compare different frame sizes and backbones and find the perfect balance between speed, classification, and detection performance. The 3D backbones are shown with the first letter of their name as a subscript for the networks (e.g. YOWO53$_{(S)}$ and

Table 1: Selected activities and the number of training videos for each

	Standing	Walking	Transporting	Hammering	Drilling	Placing/Fixing
No. of videos	95	92	21	42	27	74
No. of frames	9,145	9,163	2,840	2,442	1,840	9,981

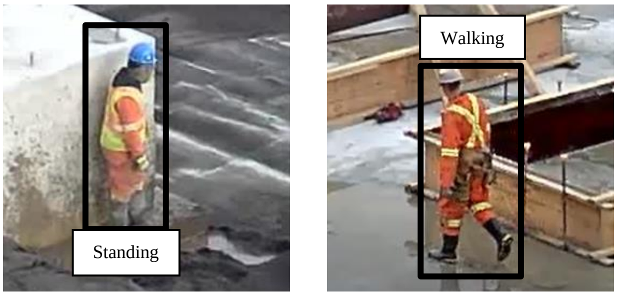
YOWO53$_{(R)}$ stand for the new version of YOWO with ShuffleNetV2 2x, and ResNext-101, respectively, as the 3D backbones).

Most of the state-of-the-art object detectors use anchor boxes instead of directly producing the bounding box locations. Anchor boxes are a set of initial bounding boxes with fixed sizes that are placed at every location of the output feature map. The network then produces a set of offsets to correct the shape and location of the boxes with the highest overlap with objects of interest so that they fall entirely inside the updated anchor boxes. The size of anchor boxes in YOWO53 is with respect to the size of the final feature map, which subsequently depends on the size of the input frame. Therefore, they are adjusted for the size of the workers in each input size.

IMPLEMENTATION AND CASE STUDY

Dataset

Table 1 shows the statistics of a subset of identified activities in the site that are used to generate the dataset for this research. An example of each activity is shown in Figure 3. The videos are collected from a construction site near Montreal, Canada. The original video frame size was 1920x1080. However, only the 1440x720 segments from the bottom-right corner of the frames are used to remove far-field activities as their detection is very challenging, and beyond the scope of this paper. Frames are extracted at 15 FPS (out of 30 FPS) from the videos as consecutive frames were highly similar. Training and testing videos have various durations from 2 to 10 seconds to cover segments of site videos that contain the activities of interest. The training dataset consists of 157 video clips (16,566 frames) and testing is done on 39 video clips (3,720 frames). Each video contains multiple activities and workers, resulting in 351 activity instances in total. Frames were extracted and annotated with bounding boxes around workers as well as their activity using LabelImg (darrenl, 2020) annotation toolbox.



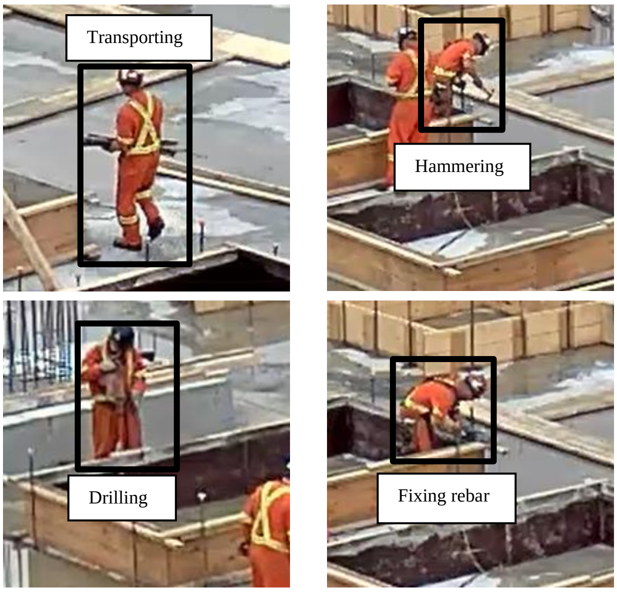


Figure 3: Examples of activities shown in Table 1

Training

The networks are trained in a Python 3.6 environment with two 32GB NVIDIA V100 GPUs. Testing is done in the same environment with a single GPU. The batch size is set to 6 for YOWO$_{(S)}$ and YOWO53$_{(S)}$, while it is set to 1 for YOWO$_{(R)}$ and YOWO53$_{(R)}$ due to memory limits. Since the dataset is relatively small, both 2D and 3D backbones are pre-trained on Kinetics (Kay et al., 2017) dataset, and frozen except for the last two layers of the 2D backbone, and the last layer of the 3D backbone. The CFAM block and the last convolutional layer are fully trained as well. The Darknet19 and Darknet53 2D backbones pre-trained weights are downloaded from Darknet official website (Redmon, 2021). All models are trained for 25 iterations and the best results are saved for evaluation.

Adjusting anchor boxes

The networks are trained and tested on different frame sizes. Therefore, to obtain more accurate results, a different set of anchor boxes are used for each of them as explained in the method. K- means clustering with IoU as the similarity measure is applied on the height and width of the training dataset bounding boxes to find the initial anchor box sizes. The height and width of the training bounding boxes are normalized by the height and width of the image. The clustering result is shown in Figure 4. The cluster centers are then multiplied by the size of the final feature map for each frame size. YOWO uses five anchor boxes following the setup in YOLOV2, while YOLOV3 uses 9. However, increasing the number of anchor boxes reduces the speed of the network. Therefore, five anchor boxes are used for all models including the ones with the YOLOV3 backbone (Darknet53). In addition, since the network only detects a single object (workers) in near and mid-field, using more variations in anchor sizes is not necessary.

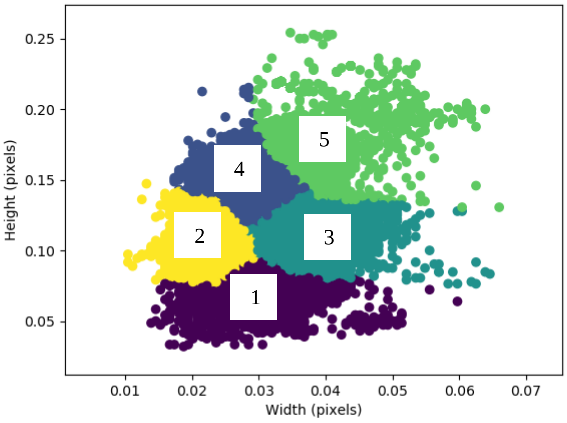


Figure 4: The clustering result of normalized height and width of bounding boxes

Classification, detection, and video-mAP

Only detections with more than 0.5 IoU with one of the ground-truths are counted as true positive (TP) for evaluation of classification accuracy, detection recall, overall precision, overall recall, and overall f1-score using (2) to (6). Overall precision and recall measure both detection, and classification performance of the framework; and are calculated for activity detections (referred to as 'Total detections' in (4)), and TPs with more than 0.25 confidence score. The confidence score is the multiplication of both detection and class confidence.

$$Classification\ accuracy = \frac{Correctly\ classified\ TPs}{Total\ TPs} \tag{2}$$

$$Detection\ recall = \frac{Total\ TPs}{Total\ groundtruths} \tag{3}$$

$$Overall\ precision = \frac{Correctly\ classified\ TPs}{Total\ detections} \tag{4}$$

$$Overall\ Recall = \frac{Correctly\ classified\ TPs}{Total\ groundtruths} \tag{5}$$

$$Overall\ f1 = 2 \times \frac{overall\ precision\ \times overall\ recall}{overall\ precision +\ overall\ recall} \tag{6}$$

Table 2 shows the classification accuracy, detection recall, and overall f1-score for different models. YOWO53$_{(S)}$ shows a great improvement over the original YOWO$_{(S)}$ in all three measures from 76.4% to 85.3%, 89.7% to 98.5%, and 0.674 to 0.823 respectively, using 896x896 input frames. The same pattern holds for smaller input sizes. Note that YOWO53$_{(S)}$ with the smallest input size gives almost the same results as the original YOWO$_{(S)}$ with the largest input size. Additionally, YOWO53$_{(R)}$ is tested with a slight improvement in both classification accuracy, detection recall, and overall f1-score (77.0%, 91.1%, 0.688 respectively) with 448x448 input frames running at 4 FPS. However, due to the high computational complexity and large size of the network, it was not possible to fit larger frames even in two GPUs for the training stage. YOWO$_{(R)}$ with 448x448 input frame resulted in much more improvement over YOWO$_{(S)}$ with 79.0%, 61.7%, and 0.49 classification accuracy, detection recall, and overall f1-score.

Video-mAP is calculated using multiple IoU thresholds, and measures the area under the precision-recall curve for activity tubes (i.e. linked detected boxes with the same activity); thus it is a good metric for evaluating the spatiotemporal detection ability of the method. The precision and recall are computed as in (4), and (5). 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 ground truth activity tube, and the activity tube recognized by the network. The resulting value is 3D IoU as shown in (7). 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 a correctly classified TP; otherwise, it is an FP. The precision and recall are calculated using these TPs and FPs. Finally, video-mAP is computed by taking the area under the precision-recall curve of activity tubes for each class and each IoU threshold.

$$3D\ IoU = \left(\frac{1}{No.boxes} \times \sum_{boxes} IoU\right) \times \frac{temporal\ intersection}{temporal\ union} \tag{7}$$

The values reported in Table 3, are the average video-mAP over 0.05, 0.1, 0.2, 0.3, 0.5, and 0.75 IoU thresholds for each class using YOWO53$_{(S)}$. The average video-mAPs over all classes are reported in the last column for each input size. Similar to Table 2, increasing the input size improves the average result; with the largest input size (896x896), giving 0.686 video-mAP. It can be noticed from Table 3, that transporting material and walking have low video-mAP. This may be explained by the fact that transporting materials is a combination of the walking activity and holding an object in hands. The two activities have similar mobility and pose. Therefore, they can be easily confused if the object is not visible while workers are facing away from the camera. Having few training videos for the transporting activity can be another explanation of low video-mAP for this class.

Table 2: Comparison of classification accuracy, detection recall, F1-score, and speed of different networks

Network	Input size	Classification accuracy (%)	Detection recall (%)	F1-score	FPS	
					Batch size 1	Maximum batch size
YOWO$_{(S)}$	896	76.4	89.7	0.674	7.9	14.4 (batch size 4)
	704	78.4	86.0	0.622	9.6	22.9 (batch size 7)
	512	74.8	54.1	0.386	11.3	48.4 (batch size 14)
	448	76.5	35.1	0.261	12.0	61.6 (batch size 14)
YOWO53$_{(S)}$	896	85.3	98.5	0.823	5.2	5.2 (batch size 1)
	704	83.1	98.4	0.795	7.4	11.1 (batch size 3)
	512	81.9	93.0	0.745	9.3	22.2 (batch size 5)
	448	76.1	89.7	0.672	9.5	29.3 (batch size 7)

Table 3: Per-class, and overall video-mAP for different input sizes with YOWO53$_{(S)}$

Input size	Standing	Walking	Transporting	Hammering	Drilling	Placing/Fixing	Average
896x896	0.759	0.467	0.289	0.777	0.999	0.823	0.686
704x704	0.722	0.412	0.290	0.611	0.999	0.883	0.653
512x512	0.693	0.404	0.333	0.778	0.833	0.821	0.643
448x448	0.672	0.331	0.332	0.778	0.833	0.783	0.621

Speed performance

The speeds of different models are shown in the last two columns of Table 2. Runtime speeds are calculated on one 32GB NVIDIA V100 GPU for both batch size 1, and the maximum batch size that fits in the GPU. The batch size is increased step by step for each frame size to find the maximum value. Smaller inputs are processed faster by the network with batch size 1; however, they also allow using bigger batch sizes resulting in even faster processing. The maximum speeds achieved by using maximum batch sizes are shown in the last column of the table. YOWO53$_{(S)}$ is slower than YOWO$_{(S)}$ on 896x896 input frames running at 5.2 maximum FPS compared to 14.4 maximum FPS. However, the improvement brought by Darknet53 allows YOWO53$_{(S)}$ to use smaller input sizes such as 448x448 running at 29.3 maximum FPS and still get the same performance as YOWO$_{(S)}$ with 896x896 input size at 14.4 maximum FPS.

SUMMARY, CONTRIBUTION, AND CONCLUSION

To address the limitations of recent computer vision-based activity recognition frameworks for construction workers, a spatiotemporal activity recognition network, YOWO, and an altered version of it introduced in this paper called YOWO53 are tested on a costume dataset containing six different activities. These networks can jointly detect and classify activities for all workers in each frame while considering the contextual information from the scene, which was not leveraged before. The input videos may contain multiple workers performing several continuous activities. The activity recognition is performed at frame-level instead of clip-level allowing a more detailed productivity analysis. The networks are jointly optimized for detection and activity recognition as opposed to the previous methods that optimized separate modules. The contribution of this paper is in applying variations of YOWO and YOWO53 for activity recognition on construction sites to test their performance based on a sensitivity analysis considering the frame sizes. YOWO53$_{(S)}$ shows better classification and detection results over YOWO$_{(S)}$ and allows using smaller input frames with real-time speed.

YOWO and YOWO53 are compared based on speed, and accuracy with different input sizes. YOWO53$_{(S)}$ with 896x896 frame size achieved higher detection recall (98.5%) and activity classification accuracy (85.3%) compared to YOWO$_{(S)}$ with a similar frame size. In addition, YOWO53$_{(S)}$ can achieve almost the same result as YOWO$_{(S)}$ with 896x896 input frames using 448x448 inputs, compensating for its lower speed. YOWO53$_{(S)}$ runs at 29.3 FPS on 448x448 input frames while the original YOWO$_{(S)}$ runs at 14.4 FPS on 896x896 frames having almost similar classification and detection performance.

REFERENCES

Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J., 1999. OPTICS: ordering points to identify the clustering structure. ACM SIGMOD Rec. 28, 49–60. https://doi.org/10.1145/304181.304187

Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., 2016. Simple Online and Realtime Tracking. 2016 IEEE Int. Conf. Image Process. ICIP 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003

darrenl, 2020. tzutalin/labelImg.

Girdhar, R., Carreira, J., Doersch, C., Zisserman, A., 2019. Video Action Transformer Network. ArXiv181202707 Cs.

Hara, K., Kataoka, H., Satoh, Y., 2018. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? ArXiv171109577 Cs.

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

Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C., 2017. Action Tubelet Detector for Spatio-Temporal Action Localization. ArXiv170501861 Cs.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A., 2017. The Kinetics Human Action Video Dataset. ArXiv170506950 Cs.

Köpüklü, O., Kose, N., Gunduz, A., Rigoll, G., 2019. Resource Efficient 3D Convolutional Neural Networks. ArXiv190402422 Cs.

Köpüklü, O., Wei, X., Rigoll, G., 2020. You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization. ArXiv191106644 Cs.

Luo, X., Li, H., Cao, D., Dai, F., Seo, J., Lee, S., 2018a. Recognizing Diverse Construction Activities in Site Images via Relevance Networks of Construction-Related Objects Detected by Convolutional Neural Networks. J. Comput. Civ. Eng. 32, 04018012. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000756

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

Luo, X., Li, H., Wang, H., Wu, Z., Dai, F., Cao, D., 2019. Vision-based detection and visualization of dynamic workspaces. Autom. Constr. 104, 1–13. https://doi.org/10.1016/j.autcon.2019.04.001

Luo, X., Li, H., Yu, Y., Zhou, C., Cao, D., 2020. Combining deep features and activity context to improve recognition of activities of workers in groups. Comput.-Aided Civ. Infrastruct. Eng. mice.12538. https://doi.org/10.1111/mice.12538

Pan, J., Chen, S., Shou, Z., Shao, J., Li, H., 2020. Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization. ArXiv200607976 Cs Eess.

Redmon, J., 2021. Darknet: Open Source Neural Networks in C [WWW Document]. Pjreddie.com. URL https://pjreddie.com/darknet/ (accessed 1.21.21).

Redmon, J., Farhadi, A., 2016. YOLO9000: Better, Faster, Stronger. ArXiv161208242 Cs.

Redmon, J., Farhadi, A., n.d. YOLOv3: An Incremental Improvement 6.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L., 2017. Temporal Segment Networks for Action Recognition in Videos. ArXiv170502953 Cs.

Yang, Xitong, Yang, Xiaodong, Liu, M.-Y., Xiao, F., Davis, L., Kautz, J., 2019. STEP: Spatio-Temporal Progressive Learning for Video Action Detection. ArXiv190409288 Cs.