Recognizing Actions in High-Resolution Low-Framerate Videos: A Feasibility Study in the Construction Sector

Benjamin Vandersmissen*, Arian Sabaghi*, Phil Reiter and Jose Oramas
University of Antwerp, Imec-IDLab, Antwerp, Belgium

**both authors had equal contribution.*

Keywords: Action Recognition, Computer Vision, Deep Learning, Video Processing.

Abstract: Action recognition addresses the automated comprehension of human actions within images or video sequences. Its applications extend across critical areas, mediating between visual perception and intelligent decision-making. However, action recognition encounters multifaceted challenges, including limited annotated data, background clutter, and varying illumination conditions. In the context of the construction sector, distinct challenges arise, requiring specialized approaches. This study investigates the applicability of established action recognition methodologies in this dynamic setting. We evaluate both sequence-based (YOWO) and frame-based (YOLOv8) approaches, considering the effect that resolution and frame rate have on performance. Additionally, we explore self-supervised learning techniques to enhance recognition accuracy. Our analysis aims to guide the development of more effective and efficient practical action recognition methods.

1 INTRODUCTION

Action recognition in computer vision automates the identification and understanding of human actions in videos and images, essential for various applications. It helps machines comprehend human movements, benefiting human-robot interaction (Rodomagoulakis et al., 2016), video retrieval (Ramezani and Yaghmaee, 2016), entertainment (Huang et al., 2017), and autonomous driving (Chen et al., 2020).

Action recognition is a field full of challenges, the most pervasive being the scarcity of annotated data, which hampers the training and generalization of models across various scenarios. Additionally, distinguishing human figures within cluttered backgrounds remains a formidable task, due to visual noise and occlusions. These challenges are exacerbated in dynamic environments like construction sites, where recognizing worker actions presents unique complexities not seen in more controlled settings typical of standard benchmarks (Fathi and Mori, 2008; Carreira and Zisserman, 2017; Feichtenhofer et al., 2017).

The relatively small scale of workers against large construction sites complicates detection and tracking, worsened by cluttered conditions and varying outdoor lighting. Additionally, video is recorded at a very low frame rate, further deviating from conventional benchmarks.

Here, we navigate the landscape of existing action recognition methodologies and assess their suitability to the construction context. We conduct this study along two distinct trajectories. Firstly, we delve into a sequence-based action recognition method , scrutinizing the efficacy of the YOWO (You Only Watch Once) approach (Köpüklü et al., 2019). Complementary to this, we explore the effectiveness of frame-based methods, the YOLO (You Only Look Once) approach (Reis et al., 2023; Redmon et al., 2016).

Through rigorous experimentation, we highlight the effectiveness of various techniques in real-world action recognition, emphasizing the importance of resolution and frame rate. Furthermore, our investigation delves into the potential enhancement through self-supervised learning techniques, providing a comprehensive understanding of their impact on action recognition performance. We expect that with this meticulous analysis we assist in paving the way for more effective and efficient action recognition methodologies in practical applications.

The rest of the paper is structured as following: in section 2 we go over related work. Our methodology — including context, datasets and evaluation metrics — is discussed in section 3. Experiments are conducted and discussed in section 4 and finally, we conclude our paper with section 5.

Vandersmissen, B., Sabaghi, A., Reiter, P. and Oramas, J.
Recognizing Actions in High-Resolution Low-Framerate Videos: A Feasibility Study in the Construction Sector.
DOI: 10.5220/0012423900003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 593-600
ISBN: 978-989-758-679-8; ISSN: 2184-4321

2 RELATED WORK

Action recognition and localization are closely related yet distinct problems with few integrated solutions. We categorize existing research into two main categories.

Action Recognition. In action recognition, 3D Convolutional Neural Networks (CNNs) such as C3D (Tran et al., 2015), P3D (Qiu et al., 2017), and R(2+1)D (Tran et al., 2018) play a crucial role, with C3D utilizing 3D convolutions and pooling layers for feature extraction, while P3D and R(2+1)D enhance efficiency through factorized convolutions. Multi-stream networks like the two-stream CNNs (Simonyan and Zisserman, 2014) and SlowFast Networks (Feichtenhofer et al., 2019) dedicate two streams to capture spatial and temporal information separately. Hybrid networks combine CNNs with recurrent layers such as LSTMs (Wang and Schmid, 2013; Kar and Prabhakaran, 2017). To integrate structural aspects, (Yan et al., 2018) employs spatio-temporal graph convolution. Similarly, (Si et al., 2019) applied GCN-LSTM with an attention mechanism to model temporal dependencies.

Action Recognition in Construction Sites. In construction sites, action recognition requires specialized methods. (Roberts and Golparvar-Fard, 2019) introduces a multi-step approach using RetinaNet for equipment detection, optical flow for consistent detection, and various techniques like SVM, HMMs, and GMMs for activity categorization. (Li and Li, 2022) encodes skeletons of workers into a feature matrix, uses an attention model in a GAN-based data imputation framework, and employes a ResNet model for action categorization. (Ishioka et al.,) adopted a 3D Inception-v1 multi-stream approach, processing RGB and optical flow frames from video clips and averaging appearance and dynamic feature maps for classification.

Action Localization. Action localization in videos draws from successful object detection methodologies like the R-CNN series (Girshick et al., 2014) which propose and classify regions in a two-stage process. For real-time applications, YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016) simplify this to a single stage, enhancing real-time performance. Innovative methods, such as those using optical flow for region proposal (Weinzaepfel et al., 2015; Zhang et al., 2020), further enhanced action localization. YOWO (You Only Watch Once) (Köpüklü et al., 2019) combined action localization and recognition in a single-stage model, using dual-branch design for comprehensive feature extraction. YOWO combines 2D and 3D feature extractors to capture spatial and spatio-temporal information. These features are then integrated through a Channel Fusion & Attention Module (CFAM) for robust action detection across frames.

The latest development in the YOLO series, YOLOv8, has found its way into recent action recognition solutions (Noor and Park, 2023; Wang et al., 2023) despite its lack of temporal information processing in videos. Various implementations reflect its versatility; (Noor and Park, 2023) applies YOLOv8 for pose estimation, whereas (Wang et al., 2023) integrates it for both action localization and recognition.

3 METHODOLOGY

In this section, we outline the methodology employed in our study, focusing on action localization within the unique context of a construction site. Subsequently, we delve into the datasets utilized for our experiments. Our objective is to investigate the effectiveness of established action recognition techniques in a scenario marked by additional challenges. To this end, we look into action recognition problem from different aspects. In doing so, we provide valuable insights for future applications in similar dynamic environments.

3.1 Data Context

One of the pivotal factors influencing the design and performance of action recognition methods is the identification of key parameters specific to each dataset. Notable disparities exist between well-established action recognition datasets (e.g., JHMDB-21 (Jhuang et al., 2013), EarthMoving (EM) (Roberts and Golparvar-Fard, 2019)) and our self-collected dataset (BoB) in construction sites, including variations in frame rate and input resolution, as outlined in Table 1. To understand the impact of these disparities on the YOWO method, we delve into a comprehensive exploration of these two dimensions (Section 4). This analysis will provide insights into how unique characteristics of our dataset may affect the performance of the YOWO method.

3.2 Input Source

While the conventional approach to action recognition relies on spatio-temporal information, frame-based action recognition can be more practical in scenarios where real-time response is crucial. As a result, we conduct an investigation into action recognition

using the BoB dataset, considering both continuous video segments and static frame inputs.

3.3 Level of Learning Supervision

In many cases, often a significant amount of unannotated data is available but only labeled data is used during training. To assess the potential benefits of unlabeled data in the BoB dataset, we first pre-train our frame-based model using self-supervised tasks, and then we conduct supervised training with action labels. This allows us to compare the performance of frame-based model pre-trained with unlabeled data against the model trained solely in a supervised manner without such pre-training.

3.4 Datasets

In addition to our proprietary construction site dataset, we incorporate publicly available datasets to scrutinize models under varying parameters such as input resolution, temporal frequency, and their combinations. This not only ensures a robust evaluation but also aligns with standard practices in the field.

JHMDB-21. The Joint-annotated Human Motion Data Base (JHMDB-21) (Jhuang et al., 2013) serves as a classical Action Recognition dataset, featuring segments dedicated to a single type of action. Notably, actions persist throughout each segment, adhering to the dataset's definition. The dataset encompasses 21 action classes revolving around human activities (Figure 2).

Earthmoving. The Earthmoving (EM) dataset (Roberts and Golparvar-Fard, 2019) is composed by videos depicting vehicles engaged in various common actions (Figure 3). Frames may exhibit multiple concurrent actions at distinct locations, and at times, no action at all. Due to its inherent complexity and direct relevance to the setting under study, it serves as our primary evaluation dataset. The dataset comprises eight distinct action classes, tailored to excavator and truck objects.

BoB Dataset. The dataset under consideration encompasses annotated videos within a construction environment with distinct characteristics from the previously mentioned settings. While the dataset contains a wide array of class labels, it follows a long-tailed distribution. Consequently, we opt to focus on a subset comprising the four most prevalent actions, collectively constituting 81.5% of all labeled actions. These actions include working, walking, standing, and flattening concrete. See Figure 1 for visual samples.

In Table 1, you can observe the distinctions between the datasets considered in this study.

Table 1: Datasets statistics.

	BoB	JHMDB-21	EarthMoving
Framerate	0.2 fps	20 fps	20 fps
Resolution (px)	3840×2160	320×240	480×272
Avg # actions / frame	4.08±2.17	1	1.72±0.45
Avg action size (px)	60×128	113×175	192×121
# Classes	5	21	7

Figure 1: Cropped 480x480 Region of Interest (ROI) from the BoB dataset.

The most significant disparities between the datasets lie in their respective frame rates and resolutions, representing the temporal and spatial axes. We delve into the influence of these factors on the overall action localization performance in section 4.

4 EXPERIMENTS

Our experiments can be divided into two phases. In the first phase (subsection 4.1 and subsection 4.2) we explore the effects of different scenarios on the YOWO model (Köpüklü et al., 2019) across two different axes, spatial resolution and temporal resolution. This serves to extract insights in the behaviour of YOWO within constrained settings. In the second phase (subsection 4.3 and subsection 4.4) we apply both sequence-based and frame-based action recognition methods on the proprietary dataset to determine the best solutions.

Mean Average Precision (MAP) is adopted as performance metric throughout our experiments.

4.1 Impact of Spatial Resolution

In construction sites, cameras with high resolution are commonly placed at a distance, which leads to a large field-of-view. As multiple actions at distance are happening within the frame, this leads to lower data resolution compared to those that characterizes standard action recognition benchmarks in the field. In this section, we will study the performance of the sequence-based model by inducing this issue manually. We conduct our experiments with both JHMDB-

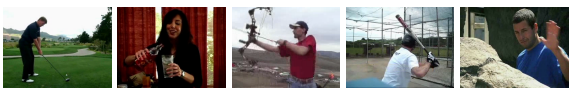

Figure 2: Frame samples from the JHMDB-21 dataset. From left to right, the classes are: 'golf', 'pour', 'shoot_bow', 'swing_baseball', 'wave'.

Figure 3: Frame samples from the EM dataset. Each sample includes a number of actions such as 'fill (truck)', 'idle (truck)', 'swing bucket', 'dump bucket' and 'load bucket'.

21 and EM, by applying varying amounts of spatial down-sampling to the input. Our findings show that small amounts of down-sampling (4× and 16×) do not significantly impact the prediction scores of the final models. However at large reductions in resolution, prediction scores are significantly impacted, albeit that some classes are more resistant than others.

Experimental Setup. The original resolutions for the JHMDB-21 and EM datasets are 320×240 and 480×272, respectively. As the YOWO model requires an input of 224×224, our first goal is to convert our data to the required resolution. Following conventional approaches, during training we use random cropping, thus using only part of the input sequence, but preserving the spatial information within the selected region, while during testing and validation we down-sample the whole sequence to the target resolution. This initial down-sampling already reduces the spatial information within the clip by 35%, respectively 62% depending on the dataset used.

To simulate scenarios with lower spatial resolutions, we applied an additional down-sampling step on both the training and validation data, by down-sampling a further 4×, 16×, 64× or 256×. This corresponds to a loss in spatial information of at least 84% and 90% for JHMDB-21 and EM, respectively.

Results. Results of the impact of spatial down-sampling can be found in Table 2 and Table 3. From these results we can conclude that in the case of JHMDB-21, a 16× down-sampling results in a small drop in the average evaluation score, while for the EM dataset this has already a drastic impact. However, we notice that certain individual classes can be less affected than others. Based on the achieved results, We find that there are broadly three different categories of actions (class labels taken from JHMDB-21):

Category I. Actions that benefit from a small down-sampling. These classes are characterised by a significant increase in mAP at 4x / 16x down-sampled compared to the baseline. Examples are 'Sit', 'Wave'.

Category II. Actions that have similar or degraded performance when down-sampling for a small amount and accelerate performance loss in the extreme down-sampling regimes. Examples are 'Catch', 'Climb Stairs', 'Jump'. Most of the classes in the studied datasets follow this pattern.

Category III. Actions that retain performance relatively well even in the extreme down-sampling regimes. Examples are 'Brush hair', 'Clap', 'Golf'.

We posit that these disparities arise due to the beneficial effects of down-sampling, which include the condensation of features. This leads to more efficient extraction of visual elements and enhanced noise reduction. This phenomenon particularly benefits classes in category I. Furthermore, certain actions exhibit exceptional distinctiveness from others in the dataset, or possess a high correlation between the action and background, allowing them to maintain their recognizability even under extreme down-sampling. Noteworthy examples include 'Golf', characterized by its distinctive motion occurring exclusively on golf terrains, and 'Pullup', distinguished by its unique vertical movement absent in other action classes in JHMDB-21.

Table 2: Average Precision results for spatial down-sampling on the Earthmoving dataset.

Setting / Class	Original	4×	16×
Idle (Excavator)	**82.89**	72.85	49.69
Swing Bucket	**83.36**	82.44	85.40
Load Bucket	**84.95**	57.34	43.17
Dump Bucket	**76.46**	72.70	58.86
Move (Truck)	84.26	**86.75**	58.86
Fill Excavator	73.12	**77.21**	52.01
Mean	**80.84**	74.89	58.00

4.2 Impact of Temporal Resolution

In this section we explore the effects of temporal resolution on the performance of the YOWO model. Typical action detection datasets such as JHMDB-21 have a high framerate, which might be infeasible to be adopted for large datasets collected in the construction space. As such, we delve into two critical temporal aspects of the YOWO model.

Firstly, we investigate temporal down-sampling, which is achieved by reducing the frame rate in a video. Additionally, as a supplementary inquiry, we explore the transferability of learned representations across different levels of temporal down-sampling.

Secondly, we scrutinize the impact of buffer size on the performance of the sequence-based model. The buffer size determines the available temporal context for the prediction of the model. While a larger buffer naturally entails a slower run-time, the relative performance gain remains a pivotal question.

Table 3: Average Precision results for Spatial and Temporal down-sampling on the JHMDB-21 dataset.

	Spatial Downsampling					Temporal Downsampling			
Setting / Class	Original	4×	16×	64×	256×	Original	2×	4×	8×
Brush Hair	87.19	83.79	**95.20**	84.77	68.75	**87.19**	76.03	86.03	74.07
Catch	**52.74**	35.64	21.77	36.56	23.72	**52.74**	28.73	17.63	18.66
Clap	88.87	**90.56**	69.24	80.68	71.78	**88.87**	77.78	74.99	81.28
Climb Stairs	82.69	**90.09**	78.70	45.41	5.69	**82.69**	65.78	64.95	54.87
Golf	99.16	**99.31**	96.32	89.71	92.23	99.16	98.11	**99.37**	96.48
Jump	25.35	14.01	**28.41**	13.21	2.72	**25.35**	13.02	9.70	10.86
Kick Ball	**78.34**	65.42	33.98	25.23	4.07	**78.34**	34.25	29.48	22.45
Pick	74.51	**78.03**	73.42	39.78	15.47	**74.51**	59.22	30.97	23.60
Pour	95.72	**97.56**	89.47	78.99	65.97	**95.72**	83.65	79.53	82.28
Pullup	99.99	98.81	**100.00**	93.52	89.09	99.99	**100.0**	**100.0**	99.98
Push	86.99	**92.65**	91.90	70.00	25.65	**86.99**	80.05	84.38	79.31
Run	37.84	41.85	**43.23**	31.19	19.48	**37.84**	27.85	13.78	21.64
Shoot Ball	**53.68**	52.05	46.76	30.54	6.71	**53.68**	40.92	43.49	21.94
Shoot Bow	**94.08**	80.11	92.13	93.87	80.99	**94.08**	81.80	82.39	86.07
Shoot Gun	79.18	**88.05**	64.81	75.79	19.41	**79.18**	49.13	62.71	47.64
Sit	39.56	42.12	**55.83**	34.47	29.13	**39.56**	38.06	29.86	18.20
Stand	36.95	40.43	40.64	**43.43**	17.10	**36.95**	29.07	16.81	30.06
Swing Baseball	63.99	53.23	**85.08**	72.89	26.04	63.99	**75.36**	64.50	49.97
Throw	43.43	**50.24**	31.12	32.05	21.57	**43.43**	33.52	15.12	17.50
Walk	54.04	**74.35**	66.41	63.22	69.11	54.04	**59.65**	59.33	50.25
Wave	39.95	**60.27**	38.60	36.93	31.70	**39.95**	29.93	18.45	30.52
Mean	67.35	**68.04**	63.91	55.82	37.45	**67.35**	56.28	51.59	48.46

4.2.1 Experimental Setup

Temporal Down-Sampling. The default buffer of YOWO is comprised of 16 frames. With k× temporal down-sampling, we sample the first frame of each set of k consecutive frames. This results in the buffer containing information spanning $16 \cdot k$ frames.

Buffer Size. We explore alternative buffer sizes to determine its impact on the inference time and performance of the resulting model. The inference time was quantified by measuring the wall-clock duration of the forward pass of the neural network. Ancillary factors, such as data loading time are not factored into this evaluation. These measurements were conducted on a system equipped with an RTX3060 GPU, an Intel Core i7 CPU, and 16 GB of RAM.

4.2.2 Results

Temporal Down-Sampling. In the analysis of the JHMDB-21 dataset (Table 3), it becomes evident that certain classes exhibit a resistance to the effects of temporal down-sampling. These classes displayed a similar resistance to spatial down-sampling, which suggests that these classes may possess inherently stronger distinguishing characteristics compared to the others. Such distinctiveness could be attributed to unique visual features, as previously noted. In all other instances, a noticeable decline in Mean Average Precision is observed with the reduction of frame rate. For the EM dataset (Table 4), we can also see some classes that are disproportionately affected by the down-sampling, while other classes suffer a much more reasonable decrease. We hypothesise that due to the temporal down-sampling the distinction between similar classes becomes too difficult to learn for the network.

Table 4: Temporal down-sampling on the EM dataset.

Setting / Class	Original	2×	4×	8×	16×
Idle (exc.)	82.89	**83.06**	27.27	33.52	3.67
Swing Bucket	**83.36**	80.07	77.82	73.49	57.35
Load Bucket	84.95	67.42	**86.34**	64.02	75.52
Dump Bucket	76.46	**76.89**	59.93	47.80	45.48
Move (truck)	84.26	84.12	**87.58**	28.44	13.58
Fill Truck	**73.12**	71.66	68.63	57.13	54.38
Mean:	**80.84**	77.20	67.93	50.73	41.66

Table 5: Buffer size effect on inference time and Mean Average Precision.

Buffer Size	Inference time	JHMDB-21	EarthMoving
8	29.38fps	47.15	**70.54**
16	20.66fps	**67.34**	80.84
32	14.64fps	54.99	**84.09**

Buffer Size. When the buffer size is reduced to 8 frames, there is a significant decrease in the mean Average Precision metric consistent across all classes in the JHMDB-21 dataset. Intriguingly, we find similar outcomes when employing a larger buffer size. In order to ascertain if these trends could be attributed to

hyperparameter choices, we conducted experiments with additional training epochs. These adjustments, however do not alter our observations.

Unlike the observations made in the JHMDB-21 dataset, we find a notable performance boost in the EM dataset when increasing the buffer size to 32 frames. This improvement, while substantial, does not exhibit the same magnitude of impact as the shift from an 8-frame buffer to 16 frames. We attribute this discrepancy to the notably longer average duration of actions in the EM dataset in comparison to those in the JHMDB-21 dataset.

In summary, our findings indicate that a larger buffer size in YOWO does not necessarily enhance performance; rather, its effectiveness is contingent upon the average duration of actions within the dataset.

4.3 Real Settings: Sequence-Based

Given the relatively small average size of actions in proportion to the frame resolution within the BoB dataset, a straightforward application of the YOWO model would be suboptimal. This arises from the necessity of resizing inputs to a 224×224 dimension, resulting in actions within the images of the BoB dataset becoming indiscernible. (Effectively, the average action would be compressed into a 4×13px window).

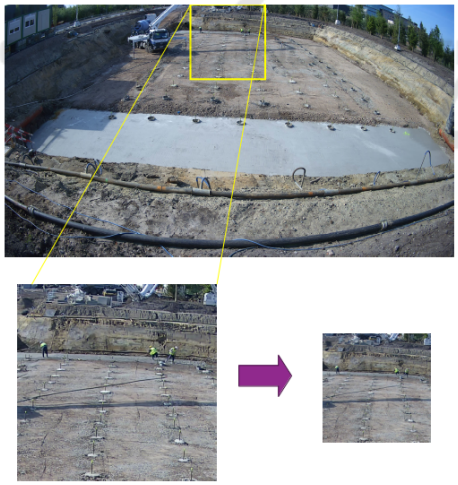

Figure 4: A schematic overview of the ROI approach.

Region of Interest (ROI) Extraction. To address this limitation, we choose to initially extract smaller candidate action regions from the image and then feed these regions through the network. This refinement enhances the spatial resolution that YOWO leverages in its processing. We will briefly explore three methods for identifying potential action regions.

Motion Analysis. In this technique, we compute the pixelwise mean of a sequence. Next, we calculate the difference between the last frame and the mean, and extract regions with high deviations.

Object Detection. This method relies on a pre-trained person detector. We generate detections and subsequently focus on regions surrounding a person.

Oracle-Based Approach. This strategy is exclusively applicable with access to the ground-truth localization boxes. In this scenario, we directly select regions that we know encompass an action.

Experimental Setup. In this section, we analyze the impact of varying ROI and buffer sizes on the performance of YOWO trained on the BoB dataset.

ROI Size. We assess three distinct ROI sizes (Table 6): 480×480, 640×640, and 800×800. During both training and validation we employ the oracle-based approach to extract ROIs. This approach is necessary during the training phase and by using it during the validation phase, we can get an upper bound on the performance of the YOWO model. The choice of ROI size presents a trade-off: smaller regions capture finer visual details but may overlook broader contextual information. Our observations listed in Table 6 affirm this intuition. Generally, smaller window sizes yield better results, as they entail less loss of pixel information during down-scaling to 224×224. Notably, the 'Walking' class stands out, performing optimally at the intermediate resolution. This exception likely arises from the added contextual information.

Table 6: Impact of the ROI size on the BoB dataset.

	480×480	640×640	800×800
Flatten Out Concrete	**70.43**	63.26	46.24
Walking	37.94	**46.61**	37.02
Standing	**57.60**	43.56	41.55
Working	**65.22**	58.45	34.71
Mean	**57.78**	52.97	39.88

Table 7: Impact of buffer size on the BoB dataset.

Average Precision	2 frames	4 frames	8 frames
Flatten Out Concrete	57.80	63.26	**64.52**
Walking	31.78	**46.61**	40.54
Standing	43.14	43.56	**45.06**
Working	62.74	58.45	**64.23**
Mean	48.87	52.97	**53.59**

Buffer Size. Following the experiments of subsection 4.2 We investigate three different buffer sizes: 2 frames, 4 frames, and 8 frames. We find that the practical difference between a buffer of 4 frames and 8 frames is marginal (Table 7). While there is a notable performance boost with more than 2 frames of buffer size, the improvement levels off. We attribute this marginal difference to the lower framerate of the BoB dataset compared to the other datasets, as addi-

Table 8: Number of samples for labeled and unlabeled sets in BoB dataset.

Datasets	Train	Valid	Test
Workers	9251	1254	2662
Background	26500	2900	None
Merged	Workers + Background		
Separated	Workers + Background		

Table 9: Performance of YOLOv8n on BoB dataset for supervised and self-supervised training.

Method	Flatten	Walking	Standing	Working	Mean
SL	0.703	**0.374**	**0.577**	**0.727**	**0.595**
CL (Separated)	**0.713**	0.309	0.508	0.717	0.565
CL (Merged)	0.700	0.322	0.496	0.717	0.429
CL (Workers)	0.587	0.281	0.377	0.599	0.461
SC (Separated)	0.516	0.22	0.241	0.533	0.377

tional temporal context may not contribute significant relevant information.

4.4 Real Settings: Frame-Based

Here we focus on action recognition based solely on a single frame. While this approach ignores the temporal information it makes real-time action recognition possible. To this end, we benefit from latest object detector in YOLO series (YOLOv8). As for action region we opt for 480×480 pixels since it yield the best results in sequence-based approach 4.3. For the extraction of the region of interest (ROI), we opt for the oracle-based approach for train/val/test splits of BoB dataset since we have access to ground-truth bounding boxes. In case of inferencing on a new data, we leverage an object detector to extract ROIs based on detected person instances. Additionally, no resize is needed to be applied since 480×480 regions can be directly fed into the model.

Experimental Setup. In this method, we operate on individual frames rather than sequences. The number of samples for labeled data (supervised training) and unlabeled data (self-supervised training) is shown in table 8.

We consider two different training strategies: *Supervised training*, which involves using data with action labels from workers to train the model in a single stage; and *Self-Supervised Pre-training*, which involves pre-training the backbone (feature extractor part of YOLOv8) and then freezing it. Subsequent layers are then trained in a supervised manner with action labels (working, walking, ...) per frame.

Adopting supervised learning techniques will allow us assess the level to which we can harness unlabeled samples in the BoB dataset. This process involves training on different subsets of data:

Workers. This group consists of data same as the labeled set, but without their associated labels.

Background. We generate random crops from each unlabeled frame of the BoB dataset. To ensure that these background crops do not contain workers, we employ a pre-trained person detector (YOLOv8).

Merged. produced by blending and intermixing the Workers and Background sets.

Separated. Here, we consider both Background and Workers categories, but refrain from mixing them together during the training process.

We also investigate two self-supervision variants: Contrastive Learning and Supervised Contrastive Learning (SupCon):

Contrastive Learning. For each batch we create a corresponding batch from a different view. Each image in the input batch (referred to as the anchor) is paired with its corresponding view as the positive pair, while all other pairs are considered negative. There is a difference in pair construction for merged and separated sets in our BoB dataset as follows. For *Separated* negative pairs are selected from the different group (Workers vs Background), while for *Merged* this constraint is not applied.

SupCon. Unlike contrastive learning, SupCon makes use of high-level labels (background vs workers). Therefore, instead of having a single positive pair for each image in the batch, all samples from the same class are considered as positive pairs.

Results. Based on the results presented in Table 9, we observe that supervised learning (SL) yields the highest performance. Contrastive learning (CL) demonstrates the second-best performance where classes are distinct, and only samples from one class are pushed away per batch. However, there is a noticeable decline in performance when employing SupCon (SC).

The preference for supervised learning stems from the robust pre-training of YOLOv8 on extensive datasets like COCO, providing a strong foundation for a wide range of vision tasks. On the other hand, when considering the unlabeled data from the BoB dataset, there exist subtle differences within each class (workers and background). This leads to only a moderate impact of self-supervised pre-training.

Regarding the performance drop observed with SupCon, we formulate two hypotheses to justify the observed results. First, the distinction between workers and background may lack a defined separation like cats and dogs. For instance, even in samples from the workers class, a good proportion of the pixels belong to the background. Second, our ultimate objective is action recognition, rather than the classification of workers versus background. As a result, the second stage of training may not align perfectly with the supervised information incorporated during pre-training.

5 CONCLUSION

Our study indicates that both spatial and temporal down-sampling generally lead to reduced performance, though spatial down-sampling shows some dataset-dependent improvements. Additionally, the effectiveness of buffer size varies with the duration of action and there is no optimal global value for that. For pre-training, we did not observe improvements from conventional self-supervision methods construction contexts. Finally, our results highlight the potential of frame-based approaches for future investigation of action recognition.

ACKNOWLEDGEMENTS

This research is part of the BoB project, an ICON project cofunded by Flanders Innovation & Entrepreneurship (VLAIO) and imec, with project no. HBC.2021.0658. The data of the construction site was provided by Willemen Groep, a Belgian construction group, with support from AICON inc.

REFERENCES

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308.

Chen, L., Ma, N., Wang, P., Li, J., Wang, P., Pang, G., and Shi, X. (2020). Survey of pedestrian action recognition techniques for autonomous driving. *Tsinghua Science and Technology*, 25(4):458–470.

Fathi, A. and Mori, G. (2008). Action recognition by learning mid-level motion features. In *CVPR*, pages 1–8.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *ICCV*.

Feichtenhofer, C., Pinz, A., and Wildes, R. P. (2017). Spatiotemporal multiplier networks for video action recognition. In *CVPR*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587.

Huang, Z., Wan, C., Probst, T., and Van Gool, L. (2017). Deep learning on lie groups for skeleton-based action recognition. In *CVPR*, pages 6099–6108.

Ishioka, H., Weng, X., Man, Y., and Kitani, K. Single camera worker detection, tracking and action recognition in con-struction site. In *ISARC*.

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J. (2013). Towards understanding action recognition. In *ICCV*, pages 3192–3199.

Kar, A. and Prabhakaran, B. (2017). A convnet-based architecture for semantic labeling of 3d lidar point clouds. In *IROS*. IEEE.

Köpüklü, O., Wei, X., and Rigoll, G. (2019). You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv:1911.06644*.

Li, Z. and Li, D. (2022). Action recognition of construction workers under occlusion. *Journal of Building Engineering*, 45:103352.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *ECCV*.

Noor, N. and Park, I. K. (2023). A lightweight skeleton-based 3d-cnn for real-time fall detection and action recognition. In *IEEE/CVF*, pages 2179–2188.

Qiu, Z., Yao, T., and Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541.

Ramezani, M. and Yaghmaee, F. (2016). A review on human action analysis in videos for retrieval applications. *Artificial Intelligence Review*, 46:485–514.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *CVPR*.

Reis, D., Kupec, J., Hong, J., and Daoudi, A. (2023). Real-time flying object detection with yolov8. *arXiv:2305.09972*.

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

Rodomagoulakis, I., Kardaris, N., Pitsikalis, V., Mavroudi, E., Katsamanis, A., Tsiami, A., and Maragos, P. (2016). Multimodal human action recognition in assistive human-robot interaction. In *ICASSP*.

Si, C., Chen, W., Wang, W., Wang, L., and Tan, T. (2019). Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *CVPR*.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *NIPS*, 27.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *CVPR*.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *PICCV*, pages 3551–3558.

Wang, Z., Liu, Y., Duan, S., and Pan, H. (2023). An efficient detection of non-standard miner behavior using improved yolov8. *Computers and Electrical Engineering*, 112:109021.

Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2015). Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Zhang, D., He, L., Tu, Z., Zhang, S., Han, F., and Yang, B. (2020). Learning motion representation for real-time spatio-temporal action localization. *Pattern Recognition*, 103:107312.