Construction Workers' Unsafe Behavior Detection through Adaptive Spatiotemporal Sampling and Optimized Attention based Video Monitoring

Mahdi Bonyani[a], Maryam Soleymani[a] and Chao Wang[b,*]

[a]Ph.D. Student, Bert S. Turner Department of Construction Management, Louisiana State University, USA
[b]Associate Professor and Graduate Program Advisor, Bert S. Turner Department of Construction Management, Louisiana State University, USA

ARTICLE INFO

Keywords:
Unsafe Behavior Detection
Construction Worker Safety
Adaptive Spatiotemporal Sampling
Attention Learning
Video Analysis
Deep Learning

ABSTRACT

In recent years, advances in construction site image analysis faced challenges, particularly in construction object detection and identifying unsafe actions. Challenges involve complex backgrounds, varying object sizes, and image quality. Existing methods address spatial and temporal features with attention mechanisms but often overlook adaptive sampling and channel-wise adjustments, missing potential spatiotemporal redundancies. This article introduces the Optimized Positioning (OP-Net) architectures and an attention-based spatiotemporal sampling approach. The OP module is introduced for object detection, which enhances channel relationships by leveraging global feature affinity associations. Additionally, we propose an innovative spatiotemporal sampling strategy that adapts to effectively identify unsafe actions in construction sites. We extensively evaluate the object detection task using the SODA dataset to showcase the efficacy and effectiveness of our approach. Furthermore, our unsafe action identification model is benchmarked on the CMA dataset, demonstrating its ability to achieve new state-of-the-art performance in accuracy while maintaining reasonable computational efficiency.

1. Introduction

Construction and civil infrastructure projects often entail complex interactions between individuals and machinery, each involving unique and dynamic processes [1]. Safety in the construction industry has been a longstanding global concern, with significantly higher rates of injuries and fatalities compared to other sectors [2]. The importance of ensuring safety at construction sites has never been more critical, especially as the industry experiences revitalization and an increased demand for infrastructure development. To illustrate, in the UK, the construction sector accounts for just 5% of the workforce but is responsible for 27% of fatal injuries, while in the US, construction accounted for nearly 20% of all occupational fatalities in 2016-2017 [3]. Despite extensive efforts to enhance safety through measures like Occupational Safety and Health Administration (OSHA) requirements, accident and mortality rates in construction have remained persistently high or reached a plateau in recent years [4]. Studies indicate that a substantial portion of construction accidents can be attributed to human factors such as unsafe behavior, lapses in supervision, and a lack of risk awareness [5]. Cognitive factors play a significant role in determining the ability of workers to promptly recognize and respond to hazards in the ever-changing environment of construction sites. Although advanced technologies like virtual reality have been explored to enhance risk awareness, they may not fully address the fundamental cognitive limitations that impact visual search and risk assessment [6].

*Corresponding author
ORCID(s):

Research indicates that a significant proportion of construction accidents, ranging from 70% to 88%, can be attributed to workers' unsafe behaviors as the immediate cause [5, 7, 8]. Studies conducted on construction sites have identified elevated rates of unsafe actions, such as the failure to use fall protection equipment, in various countries, including the US, UK, and China. Despite efforts to address these behaviors through training, their impact has been limited, with workplace negligence and oversight still contributing to a significant number of incidents [6]. For instance, it's noteworthy that 25% of these risks stem from a lack of awareness, even after workforce training efforts [9, 6]. In light of these findings, it becomes evident that alongside behavioral interventions, exploring the cognitive processes underlying risk perception, assessment, and response is crucial for improving safety outcomes.

Emerging deep learning and computer vision techniques are playing a pivotal role in complementing existing safety practices and tackling human factors within construction-related risks [10, 11, 12]. Automated monitoring systems are now capable of capturing activities across construction sites, enabling the identification of unsafe behaviors and structural defects for timely intervention. While technological advancements, such as computer vision-based behavior monitoring, offer promising avenues for enhancing risk detection and awareness, there remains a need for research to effectively integrate these findings into improved safety management and training programs [11, 12]. Leveraging large visual datasets in conjunction with deep learning has the potential to facilitate more proactive and targeted risk mitigation strategies on construction sites. In summary, adopting a multilayered approach that addresses both unsafe

behaviors and cognitive factors holds great promise in the ongoing efforts to reduce construction-related accidents.

Video analysis for ensuring construction site safety faces several challenges, primarily stemming from the redundancy between frames and the complexity of detecting crucial information. To address these issues, adaptive sampling techniques are preferred over fixed sampling rates. This preference arises from the fact that over-sampling can lead to excessive computational expenses while under-sampling risks missing critical visual details. Recent research has delved into the realm of adaptive temporal and spatial sampling, aiming to optimize the trade-offs between performance and efficiency, with a focus on specific action categories [13]. In terms of spatial sampling within individual frames, adopting an adaptive approach prevents fixed schemes from either overlooking important regions or over-processing less significant areas. The adoption of intelligent and adaptive sampling techniques is expected to play a vital role in optimizing video analysis for construction safety monitoring.

Deep convolutional neural networks (CNNs) show promise for visual analysis on construction sites, but small, dense objects in complex backgrounds pose challenges. State-of-the-art approaches leverage strategies like efficient network design, labeling techniques, and anchor generation to boost performance [14, 15, 16, 17, 18]. Attention mechanisms that adjust feature representations are also critical for handling construction site complexity. Spatial attention focuses on similarity comparisons between feature positions, enabling global context modeling [19, 20, 21, 22, 23]. Channel attention re-weights feature maps based on significance for the detection task [24, 25]. However, current channel attention methods lack relational modeling between channels that can capture feature dependencies. An optimized position attention mechanism is proposed to enhance channel attention through feature transformers, improving the handling of diverse object sizes and backgrounds.

Integrating adaptive sampling techniques and optimized position attention will enable more efficient and accurate automated visual monitoring on construction sites. Intelligent frame sampling reduces redundant computations to improve efficiency while ensuring critical information is retained. Deep CNNs with attention mechanisms can handle complex backgrounds and varied object types and sizes. Tight integration of computer vision techniques with safety management systems will enable proactive hazard identification and prevention on dynamic construction sites.

In this paper, we propose a two-part approach to identify and analyze risky behaviors in construction sites through computer vision techniques: first, we develop an object detection model trained on the SODA dataset to locate objects in videos, and second, we develop a novel spatiotemporal sampling and utilizing this detection model to identify unsafe worker actions using the CMA dataset, with the goal of improving safety management. In summary, the contributions of the proposed method are as follows:

- We propose an end-to-end network for object detection that utilizes the different proposed attention module collaborative learning approach to capture a more comprehensive feature.
- A novel and effective approach to sampling in the spatiotemporal domain is proposed that involves implementing pre-scan by temporal sampling and skipping processing and involves enhancing spatial sampling in the imagined and observed attention.
- We illustrate the performance of the proposed methods on the object detection task on SODA [26] and the unsafe behavior recognition task on CMA [27] datasets. Compared with the baseline equivalents, our method decreases computational complexity at an acceptable accuracy loss.

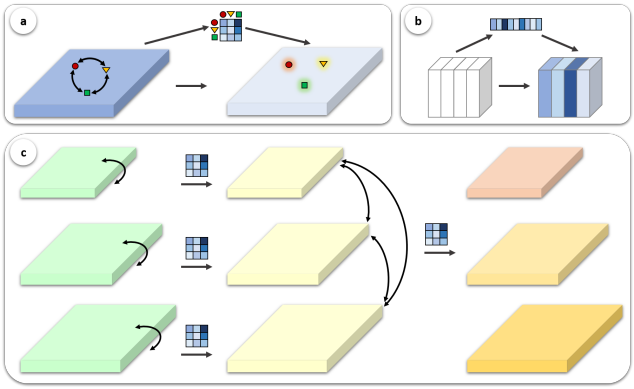

Figure 1: Schematic of the optimized-position methods. (a) Spatial attention mechanism. (b) Channel attention mechanism. (c) Our proposed optimized-position (OP).

2. Related works

Recent advancements in AI object recognition have introduced the capability to identify multiple objects, representing a significant leap compared to semantic segmentation and classification, which are limited to recognizing individual objects. However, one limitation of object recognition is its inability to precisely locate objects within an image, despite its capacity to differentiate between them. For the simultaneous identification of multiple objects and their spatial locations, object detection techniques come into play. In a study conducted by Son et al. [28], Faster-CNN with bounding boxes was employed to detect the presence of construction workers in the industry. While this classification approach can identify hazard types, it can only detect one object at a time, which restricts its direct practical application for hazard identification. Furthermore, the absence of a standardized benchmark for assessing hazard existence, coupled with a lack of indication regarding their probability levels, poses challenges. Although Luo et al. [29] used CNN to detect construction workers' activities, the study did not delve into the safety risks associated with construction.

Given the numerous safety risks on construction sites, there is a pressing need to develop object detection methods that are tailored to real-life situations. Additionally, object

detection plays a pivotal role in overlaying as-built models onto as-planned Building Information Modeling (BIM) models, enabling the quantification of work completed and the calculation of percent completion. This involves the identification of specific construction features using a dedicated BIM model for a given type of building structure, such as an RCC structure. Vision datasets are leveraged to identify various construction features and measure them in terms of quantity and relative position. Object detection methods offer a degree of automation based on the chosen technique and BIM integration, enhancing the efficiency of the process.

2.1. Detecting Unsafe Behavior through Data Sensing or Image Processing

A broad spectrum of signals originating from workers on construction sites has been collected for the purpose of identifying unsafe conditions. Li et al. [30] combined a Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) networks, utilizing acceleration signals. In another study by Bangaru et al. [31], electromyography and inertial measurements were employed to discern the activities of builders. Notably, Antwi-Afari et al. [32] demonstrated the detection of awkward working positions through the use of wearable equipment. Jung et al. [33] proposed a method for human activity classification based on sound recognition, while Lee et al. [34] developed a novel audio-based construction safety surveillance system. However, it's worth noting that the motion of workers can be affected by some wearable sensors. These existing approaches may fall short of fully capturing the features associated with unsafe conditions due to the limited dimensionality and restricted information contained in the signals. Moreover, data cleaning becomes a challenging task, given the potential interference from other signals present on construction sites, which can obscure the specific signal required for the detection of unsafe conditions.

Construction sites can quickly become hazardous when workers fail to wear the necessary personal protective equipment (PPE), leading to accidents such as fatal falls. The proper use of PPE holds the potential to prevent numerous injuries and save lives. Several studies have sought to automate the monitoring of PPE compliance in images through computer vision (CV) techniques. Fang et al. [35, 36] proposed a systematic method for evaluating whether PPE should be worn, particularly when working at heights. Nath et al. [37] employed deep learning methods to detect whether workers were wearing vests and hard helmets. Addressing the multi-class challenge of PPE recognition in the workplace, Xiong and Tang [38] introduced a pose-guided anchoring framework. To determine whether workers were wearing helmets, Yang et al. [39] utilized an approach based on CNN architecture. Chian et al. [40] developed a CV-based detection method to automatically identify missing barricades. Fang et al. [41] visually identified construction workers navigating structures and determined their relative positions using a visual model. Unlike methods that focus on recognizing actions in images, these image-based approaches emphasize the identification and localization of various objects within an image. While it's relatively straightforward to identify unsafe behavior based on the presence or absence of an object (e.g., a worker not wearing a helmet), these image-based techniques encounter challenges when dealing with complex, prolonged unsafe behaviors. For instance, in scenarios like "Fall Down," evaluating video frames temporally becomes crucial to accurately discern whether an object poses a danger, as a single frame may provide a misleading snapshot of the situation.

2.2. Detecting Unsafe Behavior through Video Analysis

Numerous studies have endeavored to identify risky activities in videos, addressing the challenges posed by the recognition of harmful actions when working with images or signals. While several deep learning methods are available for recognizing actions, two prominent categories stand out: 3D-CNN-based approaches and two-stream-based approaches. In two-stream approaches [42, 43, 44], two distinct networks are employed to process various types of data, allowing for the effective integration of diverse features with a high degree of precision. This approach proves invaluable in obtaining a wide range of features. On the other hand, 3D-CNN-based approaches [45, 46, 47] operate by extracting both temporal and spatial features in a single operation using 3D convolutional kernels, extending beyond the capabilities of 2D convolutional kernels. However, it's important to note that 3D CNN-based methods are computationally intensive, leading some researchers to propose novel convolution functions aimed at reducing computational demands [48, 49, 50]. Furthermore, a variety of modules [51, 52, 53, 54] have been developed to enhance feature extraction capabilities, and these can be directly incorporated into networks to improve their overall performance.

Numerous studies have introduced deep learning approaches for monitoring construction sites to capture and identify risky behaviors exhibited by workers and equipment. In many of these studies, workers' skeletons serve as a primary modeling component for detecting abnormal behaviors. Han and Lee's study [55] analyzed 2D human skeleton points using a stereo camera and the point coordinates of the 3D reconstructed human skeleton [56]. They constructed an action labeling model based on these skeletons. Another approach, employing skeleton-based modeling, was developed to address the challenge of occlusion in recognizing construction workers' actions [57]. Ding et al. [58] proposed a method that models skeleton points and fuses features in the temporal domain using a temporal segment Graph Convolutional Network (GCN). Additionally, a machine learning method was introduced that utilizes a depth sensor camera to identify motions as observed by the camera [59]. This approach identifies workers' actions through their skeletons rather than relying on their movements in the video. It's worth noting that first-phase skeleton extraction models can be computationally intensive and often lack parallelizability,

making their performance a critical factor in determining the ultimate output quality.

The process of identifying worker behaviors from videos has therefore become easier for researchers. An approach based on a two-stream architecture as well as a fusion approach has been proposed by Luo et al. [60] for recognizing worker actions in construction site videos. Nevertheless, it is possible to lose information when two streams are merged by using a simple fusion strategy. The work of Ding et al. [61] aimed to analyze and recognize employees' climbs and dismounts from ladders using a deep learning model based on LSTM. A video is fed into the LSTM model with 25 feature vectors, which is not a fast method, as it requires obtaining 25 feature vectors. Using deep activity features and contextual information, Luo et al. [62] developed an action recognition model that can combine deep activity features with contextual information. Using denser trajectory data, Yang et al. [63] were able to recognize the actions of workers.

Construction equipment operation status should also be monitored at all times. The construction equipment actions of excavators and trucks were recognized from videos using a 3D-CNN method developed by Jung et al. [64]. For identifying the working status of excavators, Kim et al. [65] used two layers LSTM and CNNs. According to Bügler et al. [66], productivity information regarding excavation operations was generated through the integration of two vision-based sensing methods. Recognition of worker action has been well studied, and CV techniques have been introduced into the architecture field with great ease. Nevertheless, digital twin technology is generating exponential amounts of data, and identifying unsafe behaviors is becoming increasingly challenging. Since CNNs and LSTMs are based on convolution kernels, they are unable to address such demands adequately: Convolution kernels cannot model long-range spatial as well as temporal information well because they cannot extract connections beyond the receptive field. There are certain advantages of LSTMs when it comes to sequence modeling because they have long-term memory. When there are many LSTM layers or the temporal sequence is very long, parallel computing across layers can be extremely time-consuming, leading to a very slow computation.

2.3. Points of Departure

Deep networks can be simplified in several ways, primarily by reducing complexity through techniques such as frame skipping, input region deletion, and layer omission. During the training phase, [67] suggests stochastically dropping layers as a simplification method. BlockDrop [68] and SkipNet [69], as part of their training and validation processes, propose effective layer omission strategies based on reinforcement learning. RS-Net [70] achieves complexity reduction by seamlessly integrating features across various image resolutions and being able to switch between spatial resolutions. PatchDrop [71] employs reinforcement learning to eliminate unimportant regions from input images. Time sampling is a more appropriate choice for general video analysis applications. In some cases, representing a single frame is sufficient, making time frame repetition time-consuming [72]. Given the variable speeds at which different actions occur, video processing at multiple frames per second has been utilized [49, 73]. SC-Sampler [74] and ARNet [75] handle time sampling by employing networks to pre-check features in specific scenarios. In contrast, the formulation presented in [76] underscores frame skipping based on spatial redundancy in special cases. Meanwhile, [77] focuses on considering only partial spatial information in the time domain. In this study, we explicitly model spatial-temporal sampling using human attention, with a limited number of layers for pre-scanning. Our proposed model offers the unique capability of visualizing attention and hallucinations, rendering it more understandable.

A trained model can determine regions where it considers a certain region to be "relevant" within the output by using gradient-based methods [78**?**]. In the natural language processing domain, self-attention has just recently been presented as a method of directing deep networks' attention [**?**]. Computer vision communities have been interested in such self-attention mechanisms since they allow models to focus on important regions more [79]. This attention serves as a mechanism for identifying the important spatial and temporal frames, thereby enabling adaptive spatial and temporal sampling. A more recent attention-based approach is the vision transformer, which has been adopted in several publications [**?**] [80] [81] [82, 83, 84]. The method proposed in this paper can efficiently interchange the backbone models with all CNNs that use attention since our algorithm operates on top of it. The SAN-19 [79] architecture is used in the study, and the baseline models are optimized to improve efficiency.

3. Proposed Research Methodology

In this section, we provide a comprehensive overview of the materials, methodologies, and techniques employed in our research. Our approach is divided into two fundamental parts, we begin with introducing our proposed object detection model developed and trained using the SODA dataset. This model serves as a crucial component in our pursuit of identifying and localizing objects within various contexts. Subsequently, we delve into the second part of our methodology, which involves the application of our proposed method for identifying and analyzing risky behaviors in construction environment videos. This phase utilizes the CMA dataset to address safety concerns comprehensively.

3.1. Object Detection

3.1.1. Adjustable Channel Attention

The Channel Attention (CA) module plays a crucial role in highlighting the most important channels by considering the interdependencies between them. In CA, queries (Q) are generated based on keys (K), and values (V) are derived from these queries within sets of feature maps (M). Both $\mathbf{M}'$ and $\mathbf{M}$ share similar scales with the original $\mathbf{M}$. CA can

be realized by specifying dimensions W, H, and C for each set of feature maps, denoted as $\mathbf{M} \in \mathbb{R}^{W \times H \times C}$. This can be effectively achieved by constructing a feature map using a set of features. The implementation of CA can be formulated as follows:

$$\begin{aligned} \mathrm{sim}_{i,j} &= G_{\mathrm{sim}}\left(\mathbf{q}_i, \mathbf{k}_j\right) \\ \mathbf{w}_{i,j} &= G_{\mathrm{nom}}\left(\mathrm{sim}_{i,j}\right) \\ \mathbf{M}'_i &= \sum_j G_{\mathrm{mul}}\left(\mathbf{w}_{i,j}, \mathbf{v}_j\right) \end{aligned} \tag{1}$$

Here, $\mathbf{q}_i$, $\mathbf{k}_j$, and $\mathbf{v}_j$ represent the input of CA, and $\mathrm{sim}_{i,j}$, $\mathbf{w}_{i,j}$, and $\mathbf{M}'_i$ denote similarity, weight, and output, respectively. In this context, $\mathbf{q}_i = g_q\left(\mathbf{M}_i\right) \in \mathbf{Q}$ represents the i-th query, $\mathbf{k}_j = g_k\left(\mathbf{M}_j\right) \in \mathbf{K}$ represents the second key/value pair, and $\mathbf{v}_j = g_v\left(\mathbf{M}_j\right) \in \mathbf{V}$ represents the third key/value pair. Functions $g_q(\cdot)$, $g_k(\cdot)$, and $g_v(\cdot)$ denote the transformations for channel queries, keys, and values [85, 86]. The feature maps $\mathbf{M}$ consist of two channel features: $\mathbf{M}_i$ and $\mathbf{M}_j$. G_{sim} represents the dot product similarity function, and G_{mul} denotes matrix dot multiplication. $\mathbf{M}'_i$ represents the i-th channel feature in $\mathbf{M}'$, and its response is computed based on the enumeration of all possible channels. However, despite the ability of CA to assign different weights to different channels, coarse operations (i.e., without grouped feature representations [87, 86, 88, 89]) fail to enable effective communication among all channels. Empirical evidence has demonstrated the importance of such communication in various computer vision tasks, and this limitation hinders the effective representation of features.

3.1.2. Optimized-Position (OP)

Based on global feature affinity-pairs, OP can be used to enhance feature channel relation by setting optimized weights adaptively based on the feature channel affinity pairs. Fig. 2 shows its detailed structure. By combining the multi-head representations and concatenating them with the optimized features, we produce enhanced feature maps by using a convolution layer based on the transformer mechanism. To enable richer channel feature representations, we deploy the multi-head architecture. Multi-head can provide more feature selection when extracting features in ViT [90] and DETR [85]. Using more than one head to complement features is more efficient than learning the same contents in one head. Based on their analysis work [91], important multi-head models have one or more specialized functions that are interpretable, demonstrating the need for multi-head models.

Firstly, we divide the channel dimension into P parts. Each structure in each head is an OP module (B is the batch size), based on the divided features with shape $(B, C/P, H, W)$. A similarity matrix in the form of $(B, C/P, C/P)$ exists for the n-th head module, which is expressed as follows:

$$\mathrm{sim}^n = \begin{bmatrix} w^{pC/P,pC/P} & \cdots & w^{(p+1)C/P,0} \\ \vdots & \ddots & \vdots \\ w^{0,(p+1)C/P} & \cdots & w^{(p+1)C/P,(p+1)C/P} \end{bmatrix} \tag{2}$$

In this case, every w represents a scalar of similarity that can be learned. Following the concatenation of the partial results from these head modules, we obtain the holistic output feature maps from the original feature maps with the same shape. A process similar to the one mentioned above can be described as:

$$\begin{aligned} \text{Weight} \ &: \mathbf{w}^n_{i,j} = G_{\mathrm{nom}}\left(\mathrm{sim}^n_{i,j}\right) \\ \text{Partial result} &: \mathbf{M}^n_i = \sum_j G_{\mathrm{mul}}\left(\mathbf{w}^n_{i,j}, \mathbf{v}_{j,n}\right) \\ \text{Holistic output:} &\ \mathbf{M}' = G_{\mathrm{con}}\left(\mathbf{M}^n_i\right) \end{aligned} \tag{3}$$

Here, the weights of each channel feature and its normalized version are denoted by $\mathrm{sim}^n_{i,j}$ and $\mathbf{w}^n_{i,j}$. Several channel features contribute to the calculation of the i-th channel feature. An nth head's j-th value is denoted by $\mathbf{v}_{j,n}$. Concatenating features in the channel dimension is done with G_{con}. A multi-head OP with $O\left(PC^2\right)$ computational complexity has lower computational complexity than the previous transformer-based approaches, which have $O\left(PH^2W^2\right)$ computational complexity. We propose OP implementations based on pyramid features because they offer three advantages over CA. Communication within and between feature pyramid layers is promoted by OP, whereas most previous methods capture long-range dependencies between features within and across space. In OP, features are represented in different feature spaces as a result of the multi-head structure [92, 86]. Consequently, OP can enhance the representation of features. A construction image is analyzed by OP to detect objects. In Section 4.1, OP proposes more accurate head network proposals by increasing feature pyramid representation in construction images to solve complex background and poor imaging problems. In both oriented and horizontal tasks, OP improves state-of-the-art performance dramatically (see Section 4.1). As follows are two OP implementations of a pyramid OP show with a base OP implemented on top.

3.1.3. Base OP

We can extract feature maps from arbitrary construction images using a fully convolutional network. These feature maps can be used by OP directly to adjust weights for each channel and improve communication channels. Fig. 2(b) depicts the detailed architecture of each level of the feature pyramid (i.e., feature maps with the same scale). Base OP is implemented on the basic feature maps since it is based on the basic feature maps. A base OP works on a backbone network and is a general unit. A wide range of downstream recognition tasks can be supported by this method, while other existing head-network-based methods [93, 22] are more task-specific. Section 4.1 shows the ablation experiments that demonstrate how our base OP improves feature extraction.

3.1.4. Modified Pyramid OP

There has been extensive research on the effectiveness of feature pyramids in the computer vision field [94, 95, 96]. A rearranged pyramid OP (MP-OP) is proposed here, which

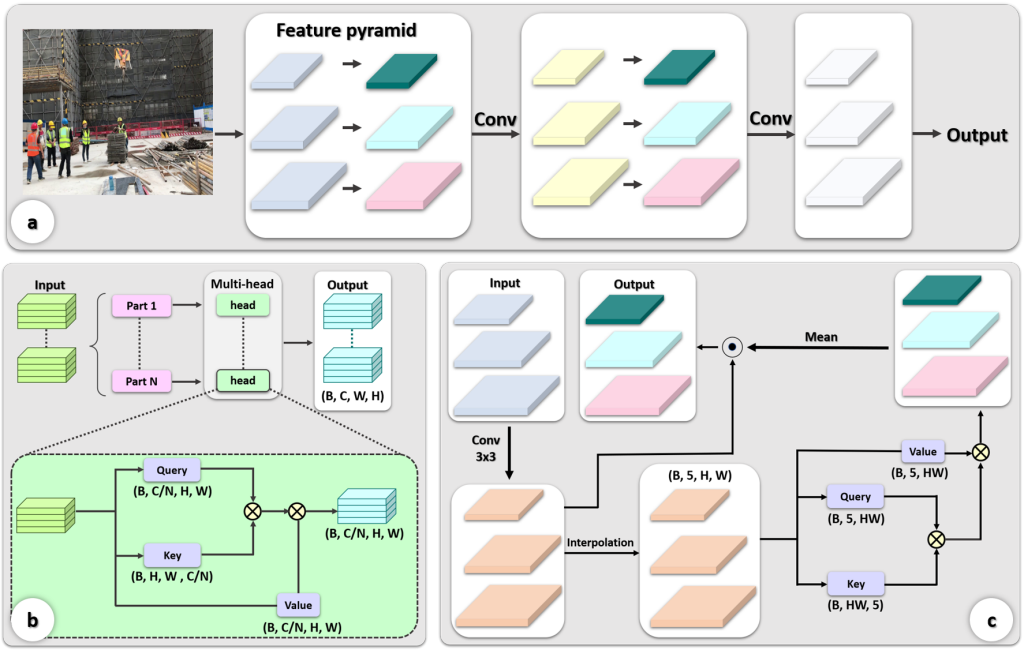

Figure 2: Our proposed OP network (OP-Net) deploys OP both on intralayer feature maps and on feature pyramids. Subfigure (a) shows the overview of the architecture of the OP-Net. Subfigure (b) shows the detailed layout of every feature pyramid. Subfigure (c) shows the overview of the architecture of the Modified Pyramid OP (MP-OP). Feature maps that utilize the OP-MP module have better representation capabilities

shows how to implement our OP on a pyramid feature [97, 98, 99]. We have developed an efficient, low-complexity, and more parameter-light MP-OP, which is applied to the in-network feature pyramid (see Section 4.1 for details). A structure of MP-OP modules is shown in Fig. 2(c). These modules extract feature pyramids from the feature pyramid network [94]. A level in the feature pyramid is viewed as a small piece of the input image's features, i.e., only a fraction of the input image's features are captured at each level. The combination of global and local information is crucial in feature extraction in order to highlight the most suitable feature in the channel dimension. Based on works [94, 96], MP-OP is used to weight different features across pyramid levels $\mathbf{M}_{S2-S6}$. The feature pyramid is illustrated in Fig. 2(c) with OP applied between the five levels in order to fully convey the information of each level. We begin by reducing the channel dimension and activating interpolation on pyramid features $\mathbf{M}_{S2-S6}$ to produce features of the same scale (same scale as S2), and then we concatenate them into $\overline{\mathbf{M}}_{S2-S6}$, which is expressed as:

$$\overline{\mathbf{M}}_{S2-S6} = G_{\text{intp}}\left(\mathbf{M}_{S2-S6}\right) \tag{4}$$

An interpolation function G_{intp} is used to reduce channel dimensions and scale. Output feature $\overline{\mathbf{M}}_{S2-S6}$ has a shape of $\left(B, 5, H_{s2}, W_{s2}\right)$. After that, MP-OP produces $\overline{\mathbf{M}}_i'$ by learning the weight between the query and the key from input $\mathbf{q}_i, \mathbf{k}_j$, and $\mathbf{v}_j$. Interactions can be expressed as

$$\begin{aligned}
\text{Input} &: \mathbf{M}_{S2-S6} \\
\text{Interpolation} &: \overline{\mathbf{M}}_{S2-S6} \\
\text{Extraction} &: \mathbf{q}_i, \mathbf{k}_j, \mathbf{v}_j \\
\text{Similarity} &: \text{sim}_{i,j} = G_{\text{sim}}\left(\mathbf{q}_i, \mathbf{k}_j\right) \\
\text{Weight:} &\ \mathbf{w}_{i,j} = G_{\text{nom}}\left(\text{sim}_{i,j}\right) \\
\text{Output} &: \overline{\mathbf{M}}_i' = \sum_j G_{\text{mul}}\left(\mathbf{w}_{i,j}, \mathbf{v}_j\right) \\
\text{Holistic output} &: \overline{\mathbf{M}}_{S2-S6}^{\text{rpcg}} = G_{\text{con}}\left(\overline{\mathbf{M}}_i'\right)
\end{aligned} \tag{5}$$

A feature $\overline{\mathbf{M}}_i'$ in $\overline{\mathbf{M}}_{S2-S6}^{\text{rpcg}}$ is at the ith level, and $\left(B, 5, H_{s2}, W_{s2}\right)$ is the shape of $\overline{\mathbf{M}}_{S2-S6}^{\text{rpcg}}$. Pyramid features are fully realized $\overline{\mathbf{M}}_{S2-S6}^{\text{rpcg}}$, but we need to find a way to feedback to them.

Furthermore, a number of methods have been used in visual recognition to verify the effectiveness of local and global information combined, and our method is a global approach. As a result, we chose to combine our MP-OP method with the existing local channel attention method. We choose classic channel attention [100] for this study. Hence, our proposed MP-OP module has the structure as:

$$\begin{aligned}
\text{Weight} &: \overline{\mathbf{M}}_{S2-S6}^{\text{avg (rpcg)}} = G_{\text{avg}}\left(\overline{\mathbf{M}}_{S2-S6}^{\text{rpcg}}\right) \\
\text{Scale} &: \overline{\mathbf{M}}_{S2-S6}' = \overline{\mathbf{M}}_{S2-S6}^{\text{avg (rpcg)}} \otimes \mathbf{M}_{S2-S6} \\
\text{Output} &: \overline{\mathbf{M}}_{S2-S6}^{\text{out}} = G_{\text{conv}}\left(\overline{\mathbf{M}}_{S2-S6}' \oplus \mathbf{M}_{S2-S6}\right).
\end{aligned} \tag{6}$$

$S2$ through $S6$ are the five outputs from $\overline{\mathbf{M}}_{S2-S6}$. A weighted average of $\overline{\mathbf{M}}_{S2-S6}$ is called $\overline{\mathbf{M}}^{\text{rpcg}}_{S2-S6}$. For each pyramid's level, the mean value is used as the weighting parameter, which is then resized to the same scale as the original level feature used by $G_{(\text{avg})}$ to distinguish between scales. A matrix cross multiplication is performed by $\otimes$, and a channel concatenation is performed by $\bigoplus$. As with the original feature pyramid, $\overline{\mathbf{M}}'_{S2-S6}$ is adjusted to the same size. In order to restore the original size of the channel, we get the output $\overline{\mathbf{M}}^{\text{out}}_{S2-S6}$ from convolution G_{conv}.

3.1.5. Network Architecture

For construction image object detection tasks, OP can boost the model's ability to learn richer communication information among feature channels. The purpose of this article is to develop an OP-Net able to detect oriented and horizontal objects in construction images. In Fig. 2, you can see the overall architecture. We propose OP-Net as a method for transforming pyramid features based on OP and MP-OP as shown in Fig. 2. A ResNet [101]-based backbone is deployed on top of [18], which has been pre-trained on ImageNet [102]. Our feature pyramid is then produced based on the feature pyramid network [94]. Our feature pyramid begins by applying base OP to each of the feature maps. A new feature pyramid is then created in which local and global communication is realized. A 3x 3 convolution reduces the dimension of the concatenated features maps to 256 channels by concatenating the original feature maps and adjusted ones together. To detect horizontal objects, we use a standard faster R-CNN [103], which is derived from the head network of the RoI transformer [18].

3.1.6. Dataset

As part of our experiments, we select a challenging dataset that is a large-scale dataset from the SODA dataset [26]. Object detection can be done both horizontally and orientedly with SODA.

It includes oriented and horizontal bounding boxes and is one of the largest construction image datasets for object detection. 188,282 objects are annotated in 2806 construction site images, which were captured by different platforms and sensors, and 15 object categories are common across all the images in the SODA database like Slogan (SL), Fence (FE), Hook (HK), Hopper (HO), Electric Box (EB), Cutter (CU), Handcart (HA), Scaffold (SC), Brick (BR), Rebar (RE), Wood (WO), Board (BO), Helmet (HE), Vest (VE), and Person (PE). The images are larger than 1920×1080 pixels and contain a wide variety of objects oriented differently and captured in different scales. Our classification consists of randomly dividing the original images into three sets: training, testing, and validation, according to the method described in [26].

3.1.7. Training Setup

In this study, we are using Faster R-CNN [103], which is combined with ResNet-101 [101] as our backbone. A feature pyramid with predefined anchors for pyramid levels S2-S6 is constructed using the FPN [94] as a neck network. A rotated head network, RoI-transformer [18], is used in oriented object detection to convert horizontal proposals into rotated ones. As outlined in [18, 26], all experimental settings and parameters are strictly consistent. End-to-end training is applied to the entire network. It is necessary for the fairness of comparisons to adjust hyperparameters even though it is conducive to further improving model performance. We set anchor size as follows in [18] and [14], for SODA, with aspect ratios of [1/2,1,2] and anchor strides of [4,8,16,32,64] at each pyramid scale. Our ablation studies are based on SODA, which does not include any data augmentation. This allows for a fair comparison and trial of the proposed method. We only add random rotation to augmentations like [18], [14], and [22] compared to SOTA methods on SODA. The number of multi-heads in the base OP can be determined by P, which is a hyperparameter for multi-head. If P is high, the dividing feature reduces the ability of the channel to relationship with each other. We set S to two in the final network based on the parameter settings of previous work [96].

As a result of our study, the learning rate is initially 0.005 and the SGD optimizer performs a weight decay of 0.0001 and momentum decay of 0.95. We set the training epoch for SODA to 80. Neither multiscale input nor TTA are used in the testing step. Furthermore, the Colab GPU is used for the experiments. According to [26], the model is evaluated and the result distribution is analyzed using the mean average precision (mAP) of each category and overall. Moreover, $GFLOPs/FPS$ and model parameters (#Params) are employed for verifying model efficiency, which is used for determining model computational complexity and runtime efficiency.

3.2. Detecting Unsafe Behavior

$\mathcal{D} = \left\{\left(\mathbf{v}_n, \mathbf{y}_n\right)\right\}_{n=1}^{N}$ is a video dataset composed of $\mathbf{v}_n$ video sequences and $\mathbf{y}_n$ ground truth labels. T frames are assumed to be of the same length in all video sequences, i.e., $\mathbf{v}_n = \left[\mathbf{f}_n^{(1)}, \mathbf{f}_n^{(2)}, \ldots, \mathbf{f}_n^{(T)}\right]$, where each frame is $\mathbf{f}_n^{(t)} \in \mathbb{R}^{H\times W\times 3}, \forall t \in \{1, \ldots, T\}$. Consider a classifier $F\left(\mathbf{v}_n\right) = \hat{\mathbf{y}}_n$ for video with $\mathcal{O}_F$ complexity. In an effort to reduce the complexity of classifier F while maintaining accuracy, another classifier $\tilde{F}$ will be constructed. To resolve this issue, we introduce a spatial sampler $\mathcal{S}$ as well as temporal sampler $\mathcal{T}$ like $\tilde{F}\left(\mathbf{f}_n; \mathcal{T}, \mathcal{S}\right) = \hat{\mathbf{y}}_n$, such that $\mathcal{O}_{\tilde{F}} < \mathcal{O}_F$. Spatial sampling picks the top-k regions according to attention map activation. By comparing each frame's attention with the model's future prediction, the temporal sampler decides whether to skip it.

3.2.1. Cumulative Global Attention

In this paper, we use a cumulative global attention formulation derived from Zhao et al. [79]. Pairwise attention is rewritten as

$$\mathbf{z}_i = \sum_{j\in\mathcal{R}(i)} \alpha\left(Q\left(\mathbf{f}_i\right), K\left(\mathbf{f}_j\right)\right) \odot V\left(\mathbf{f}_j\right) \tag{7}$$

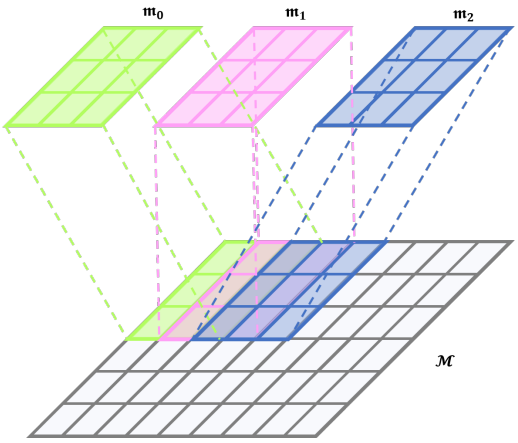

Figure 3: Cumulative attention processes take into account all the relevant information when extracting features by the sliding effects. The adjacent contexts of the green ($\mathbf{m}_0$), red ($\mathbf{m}_1$), and blue ($\mathbf{m}_2$) squares represent local attentions. The cumulative global attention $\mathcal{M}$ is represented by the bottom grid.

A spatial indices is defined as $i, j \in \mathbb{R}^2$, a key encoding is defined as $Q\left(\mathbf{f}_i\right)$, a key encoding is defined as $K\left(\mathbf{f}_j\right)$, a value encoding as $V\left(\mathbf{f}_j\right)$, and α compatibility function is defined as a softmax. The footprint $\mathcal{R}(i)$ defines such compatibility functions locally. The local attention at i can then be expressed as follows:

$$\mathbf{m}_i = \left[\alpha\left(Q\left(\mathbf{f}_i\right), K\left(\mathbf{f}_j\right)\right)\right], j \in \mathcal{R}(i) \tag{8}$$

As well as modeling the underlying relationship between neighboring footprints, we must also learn to generate such overlapping attention map, e.g., $\mathbf{m}_i$ and $\mathbf{m}_{i+1}$ overlap. As a result, a global attention map can be generated more effectively if the contexts are already encoded. To calculate it, we used the following formula:

$$\mathcal{M} = \sum_i \mathbf{m}_i \otimes \mathbb{1}\{\mathcal{R}(i)\} \tag{9}$$

In this case, $\mathbb{1}\{\mathcal{R}(i)\}$ represents a function that deletes locations externally of contexts $\mathcal{R}(i)$, as well as $\otimes$ represents the product of $\mathbf{a}_i$ and the contexts associated with it. It is important to note that $\mathbf{m}_i$ and $\mathcal{R}(i)$ have the same dimension in the spatial domain, while $\mathcal{M}$ as well as $\phi(\mathbf{f})$ (the input feature map) have the same spatial dimension. We use "attention" throughout the remaining sections to mean the cumulative global attention unless stated otherwise.

The kernel window sliding concept is similar to that of neighboring contexts. Overlapping regions will occur if the kernel size is larger than the stride. Fig. 3 illustrates an example where the kernel size is three and stride is one, resulting in an overlap of two. Due to the duplicated information in overlapping regions, representing attention as $\mathbf{m}_i$ is not efficient. The imagination then learns to generate its future version by averaging them as $\mathcal{M}$'s. In the updated manuscript, we have clarified this point.

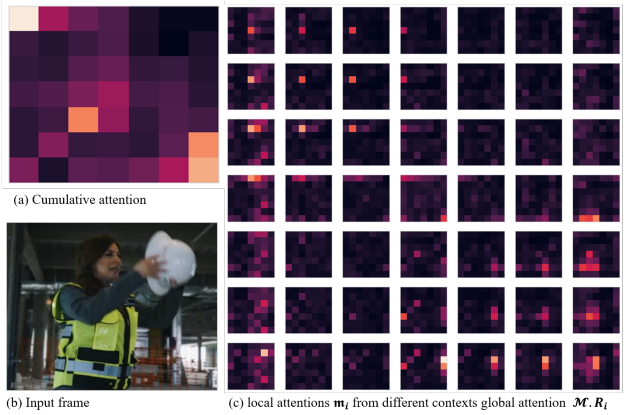

Figure 4: Cumulative attention and local attention are extracted from the same layer as well as the same input frame.

As shown in Fig. 4a, given the same input as Fig. 4b, local attentions are combined to create cumulative global attention over multiple contexts (Fig. 4c). In this example, the neighboring $\mathbf{m}_i$ overlap with each other, similar to convolution (due to stride is 1). It is easier to learn when we use $\mathcal{M}$ since we don't need to encode such overlapping conditions. The worker and the helmet (bottom-left corner) appear to be the "important regions" of the input frames that are activated by $\mathcal{M}$. To find the region of interest in the proposed spatial sampler used from global attention maps.

3.2.2. Space Domain Sampler

In humans, foveal vision is a high-resolution input that is provided at the exact location where it matters. The spatial sampler is similar to a fovea in humans. Informally, we rescale (using the down-sampling weigh d) and crop $\mathbf{f}$ $\left(W' < W, H' < H\right)$ at k various locations to obtain the associated high-res and low-res frames, $\mathbf{f}_{h,k} \in \mathbb{R}^{H' \times W' \times 3}$ as well as $\mathbf{f}_l \in \mathbb{R}^{\frac{H}{d} \times \frac{W}{d} \times 3}$, respectively. As a result, our hyper-parameter d determines where to crop $\mathbf{f}_{h,k}$, while our spatial sampler $\mathcal{S}$ determines how big the cropping regions should be. Selecting the regions with the highest summation based on the entire $\mathcal{M}$, we look for all regions that are connected. On the basis of the spatial dimension scaling between $\mathbf{f}$ and $\mathcal{M}$. Then, we return those areas to pixel space using linear projection.

The spatial sampler is shown in Fig. 6. The top-k areas of the raw frame $\mathbf{f}$ are sampled using the attention extracted from the low-res frame $\mathbf{f}_l$. Thus, $\mathbf{f}_{h,k}$ maintains the original resolution of $\mathbf{f}$, while having a lower spatial dimension. The global average pooling layer was applied to remove the spatial domain after the feature extractor due to the proposed method utilizing the same architecture to process frames of various resolutions. The three-head GRU was applied to process the features as the classifier. A strong learning feature can be encouraged at each resolution by concatenating low-res and high-resolution features. Our constraint is to make

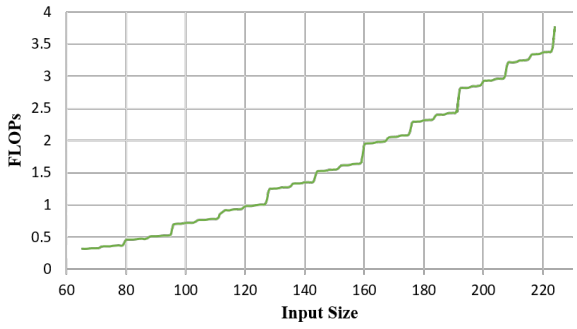

Figure 5: Effect of the various input size on SAN19's complexity.

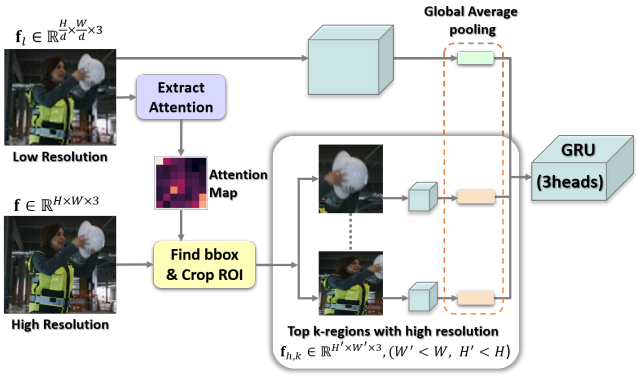

Figure 6: Overall architecture of spatial sampler. The top-k areas of the raw frame $\mathbf{f}$ are sampled using the attention extracted from the low-res frame $\mathbf{f}_l$. The spatial dimension of the features is eliminated during the final global average pooling process before they are pooled and sent into the three-head GRU classifier. Strong learning features at every resolution are promoted by the heads, which correlate to high-res, low-res, as well as their concatenation features.

$\mathbf{f}_l$ and $\mathbf{f}_{h,k}$'s less complex than $\mathbf{f}$'s in terms of scaling weigh d as well as bounding box size W', H'. According to our complexity analysis in Fig 5, we decide on d = 2 and $H' = W' = 64$.

3.2.3. Imaginary

Predicting future information is the objective of our imaginary component. We do not need to run any further inferences if our prediction matches the actual future observation. In our imaginary component, just the activation map of attention is generated instead of the entire RGB frame. The generation of these attention maps is easier than creating RGB frames, since they can locate important regions of the inputs. $\boldsymbol{\mathcal{M}}^{(t+1)}$ have equal attentions $\boldsymbol{\mathcal{M}}^{(t)}$ if the frame changes are small then It's moving slowly sufficiently, thereby imaginary component the future attention from the current attention if the temporal consistency is assumed. The formal definition of $\mathcal{J}$ is as follows:

$$\tilde{\mathcal{M}}^{(t+1)} = \mathcal{J}\left(\mathcal{M}^{(t)}\right) \tag{10}$$

such that $\tilde{\mathcal{M}}^{(t+1)} \approx \mathcal{M}^{(t+1)}$

The imagination is called $\tilde{\mathcal{M}}^{(t+1)}$ (predicted attention in the future). By comparing the structures of input tensors, we use the SSIM [73] to measure the similarity between $\tilde{\mathcal{M}}^{(t+1)}$ as well as $\boldsymbol{\mathcal{M}}^{(t+1)}$. During training, we minimize the loss of belief in the imaginary:

$$\ell_b = -\frac{1}{T-1}\sum_{t=2}^{T} \text{SSIM}\left(\mathcal{J}\left(\mathcal{M}^{(t-1)}\right), \mathcal{M}^{(t)}\right) \tag{11}$$

SSIM() measures the similarity of the imaginary $\mathcal{J}\left(\mathcal{M}^{(t-1)}\right)$ to the attention $\mathcal{M}^{(t)}$, providing an indication of the degree to which the two images match or diverge. Negative SSIM scores are minimized because the value of SSIM is between 0 and 1, SSIM with a larger value indicating an increased similarity.

As part of the training routine, we employ the teacher forcing technique [104] and create a CNN-LSTM [105] with two phases of encoder and decoder layers. As the number of imagined frames increases, the trick is to gradually increase the input. Our imagination undergoes a rewrite during training as follows:

$$\tilde{\mathcal{M}}^{(t+1)} = \begin{cases} \mathcal{J}\left(\mathcal{M}^{(t)}\right), & p \le F_r \\ \mathcal{J}\left(\tilde{\mathcal{M}}^{(t)}\right), & p > F_r \end{cases} \tag{12}$$

A uniformly randomized population is represented by p and a teacher forcing ratio by $F_r \in [0, 1]$. The imagination probability from $\tilde{\boldsymbol{\mathcal{M}}}^{(t)}$ is increased by starting with a ratio of 1 and gradually decaying with time. There is no evaluation for F_r, as well as $\tilde{\mathcal{M}}^{(t+1)}$ equals $\mathcal{J}\left(\tilde{\mathcal{M}}^{(t)}\right)$. Imaginaries' hidden memories are initially triggered during both training and evaluation phases like $\left(\widetilde{\boldsymbol{\mathcal{M}}}^{(t+1)} = \mathcal{H}\left(\mathcal{M}^{(t)}\right), \forall t \le t_{\text{warm}}\right)$.

The Fig 7 illustrates an example of an imaginary resulting from a sequence of frames. The input frame is shown in the first row of the figure, while the second row shows the attention extracted from a specific layer. Finally, the third row of the figure showcases the imaginary element that is created through our specific process. As our attention is generated in the future, imagination is missing in the first frame. Observations indicate that the two hands are the most active areas of attention here. Similarly, in both attention and imagination, these regions move in time with the hands. Consequently, our imagination predicts the future locations of important regions. A comparison of the structural similarities between imagination and attention is also provided at the bottom using the negative SSIM scores. We are only using the imagination for the temporal sampler's reference, not to generate a perfect one.

3.2.4. Time Domain Sampler

A video sequence $\mathbf{v} = \left[\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(T)}\right]$ is adaptively selected to represent $\mathbf{v}$ by selectively selecting the most relevant frames based on the temporal sampler. This is close to how humans process video sequences prior to paying

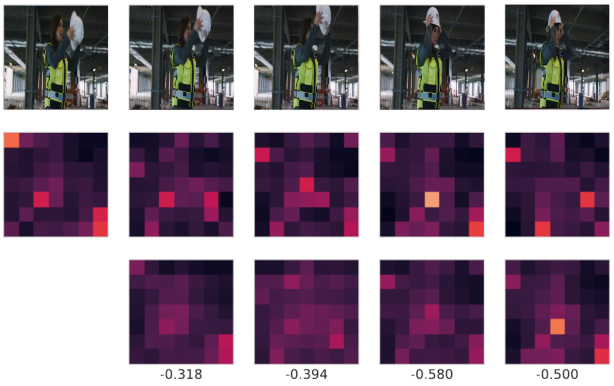

Figure 7: The input frame's attention and associated imagination. The lowest SSIM ratings for both imagination and attention are negative (0 indicates most dissimilar and -1 indicates most similar). The worker's movements are consistent across input frames in parts of the imagination and attention that are activated, demonstrating the time-related feature.

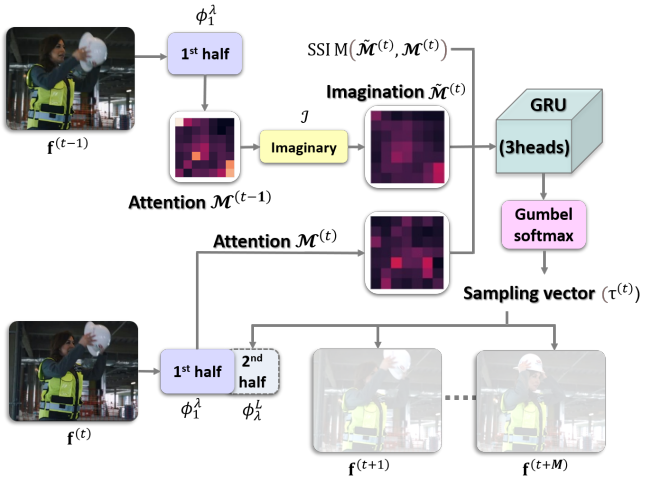

Figure 8: Overall architecture of temporal sampler. A GRU is supplied the first half of the model's attention at time t, the imagination generated at time $t-1$, as well as their SSIM score to determine the number of frames to ignore in order to compute the sampling vector $\tau^{(t)}$. Between frames, model weights are transferred.

attention. When we can reasonably predict a frame's attention $\mathbf{f}^{(t)}$, it is considered unimportant. From Section 3.2.1, a model of layers can be used to retrieve imaginary as well as attention at any layer. A temporal sampler may choose, as the temporal sampler does, as long as attention is captured at layer $\lambda < L$, the last $L-\lambda$ layers can be skipped. The model can be run adaptively from layer λ.

Considering a deep network of L layers in terms of its feature extractor, we are able to divide it into two components at layer $\lambda \in \{1,\dots,L\}$, like $\phi_1^L(\mathbf{f}) = \phi_\lambda^L\left(\phi_1^\lambda(\mathbf{f})\right)$. Pre-scanning is done using the first half ϕ_1^λ, while classification can be done later utilizing the second half ϕ_λ^L, augmented with other modalities. A temporal sampler $\mathcal{T}$ creates the sampling routine through the computation of $\tau = \left[\tau^{(1)},\dots,\tau^{(T)}\right]$, like $\mathcal{T}(\mathbf{v}) = \left[\mathbf{f}^{(t)} \times \tau^{(t)}\right]_{t=1}^{T}$ with $\tau^{(t)} \in \{0,1\}^{M+1}$, where $\tau^{(t)}[m] = 1$ represents the possibility of skipping m frames. The temporal sampler is shown in Fig. 8. A feature extractor ϕ_1^λ is used to extract the attention $\mathcal{M}^{(t)}$ at time t. An $\tau^{(t)}$ sampling vector is generated by concatenating the feature with the SSIM score and imaginary. Gumbel Softmax [106] is used to transform the output features into differentiable sampling vectors.

In order to skip frames, we use the sampling vector as a starting point. In the case of $m^* = \text{argmax}_m\, \tau^{(t)}[m]$, where $m^* = 0$ and $m^* \in [1, M]$ denote the number of skipping frames, there are two possible outcomes. For instance, the first case skips nothing and continues running the rest of the network. Therefore, it is the complexity of the full pipeline $\mathcal{O}_{\text{full}}$. By skipping computation on the subsequent frames, the second mood also pre-scans the current image. As a result, recurrent models are able to propagate classification results and memory. In this case, $\mathcal{O}_{\text{pre}} = \mathcal{O}_{\phi_1^\lambda} + \mathcal{O}_H + \mathcal{O}_\mathcal{T}$, where $\mathcal{O}_{\phi_1^\lambda}$ denotes first half, $\mathcal{O}_H$ denotes hallucinator, and $\mathcal{O}_\mathcal{T}$ denotes temporal sampler. $\mathcal{O}_{\text{pre}}$ is the complexity of running a classifier, spatial sampler, as well as other modalities, while $\mathcal{O}_{\text{rest}}$ is the complexity of operating the pipeline's remaining sections. Our training policy uses the following weighted sum reliability loss $\left(\ell_r\right)$ as well as $\ell_{\text{class}} \cdot \ell_r$ define as:

$$\ell_r = n_{\text{pre}} \cdot \mathcal{O}_{\text{pre}} + n_{\text{full}} \cdot \mathcal{O}_{\text{full}}\,, \tag{13}$$

A full inference frame consists of n_{full}, while a pre-scanning frame consists of n_{pre}. Part two of the pipeline, $\text{argmax}_m\, \tau^{(t)}[m] \neq 0, \forall t$, might not get any frames if there are no constraints. The pipeline is run fully at the beginning of the frame to prevent this scenario. Also, it ensures that we have at least one frame's classification result, which helps initialize memory for recurrent models.

3.3. System Performance Evaluation

The final evaluation of our system is conducted on CMA [27], using the training and validation splits of [27]. By using the CMA dataset, the proposed method can be trained to predict unsafe behavior actions, thus enhancing safety management in construction workplaces. For the dataset, there are seven categories of worker behavior on construction sites, with 105 to 365 clips for each category, resulting in 1595 samples altogether. These actions can be utilized to train a deep learning model to detect unsafe behaviors such as distraction, improper use of personal protective equipment, and safety performance degradation. For instance, the "Talk" and "Smoke" actions can be associated with worker distraction and potential safety hazards, while the "Lack of Safety Equipment" action can indicate improper use of personal protective equipment. Every clip lasts 1-9 seconds and has a resolution of $400 \times 300 - 1920 \times 1080$. The frames per second of every video clip are fixed at 30 and the clips contain only one action. Due to ℓ and samplers are

based on attention from vision information, RGB inputs are used as the guiding modality of the system. The FLOPS per frame are also reported for the system's efficiency. FLOPS are proportional to time and energy consumed during inference. Accumulated FLOPS as well as average FLOPS per frame used for the experiments with a temporal sampler since model complexity varies over time. Furthermore, we report the tradeoff criteria, which are expressed as GFLOPS per accuracy, for comparing effectiveness among different models. An average accuracy of one percent is measured by the amount of computation required.

3.4. System Implementation

We extract features using SAN19 as the backbone (equivalent to ResNet50) with pairwise self-attention [79]. The attention maps in Fig. 9 are based on the bottleneck layers of SAN19, with inputs sized $112 \times 112 \times 3$, positioned in the top-right corner. It is evident that the later layers result in attention maps that are smaller and more concise, which suggests that imagination of the future is easier. Fig. 10 depicts the names of bottleneck layers on the horizontal axis. The vertical axis showcases the accumulation of FLOPs as more layers are added. This demonstrates how FLOPs increase with the inclusion of additional layers in the model. Our experiments show an appropriate balance between performance and complexity at layer3-0 of SAN19, so we extract attention from there with a dimensionality of 7 x 7 x 32. Conv-LSTM with 32 hidden dimensions is the imaginary. The encoder and decoder use a 3 x 3 and 32 channel kernels for the 2D conversion layer. This action classifier uses a three-head GRU for the cropped high-res RGB (local) as well as low-res RGB (global) features, which are then joined together by the primary GRU. In order to improve feature extraction, the multiple heads design focuses the network's attention on the prominent features in the cropped regions, minimizing the network's dependency on the low-resolution input. We use the same 2 layers and 1024 hidden dimensions for the temporal sampler as well as the GRU classifier.

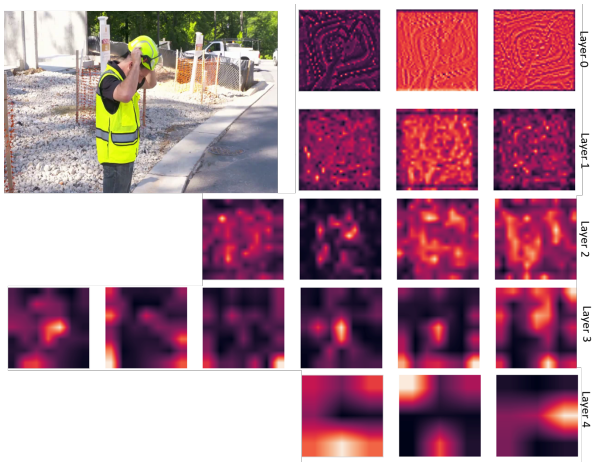

Figure 9: SAN19 is Used to extract attention. Located in the top-left corner of the figure is the input frame. There is a disparity between the earlier and later layers of attention, which indicates more fragmented regions. A bi-linear interpolation method is used to visualize the attention maps across different layers to improve visibility. The reason for the lack of normalization in the color mapping of the visualization is the variation in value ranges across different layers.

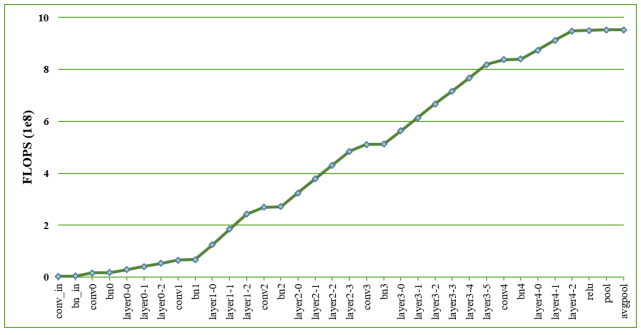

Figure 10: Accumulated complexity is achieved through a combination of various layers of the SAN19 model with a size of $112 \times 112 \times 3$, each one contributing to the overall architecture of the model. This complexity allows for detailed insights into the underlying structure and dynamics of the SAN19 model.

3.5. Dataset

By using the CMA dataset, deep learning algorithms can be trained to predict unsafe behavior actions, thus enhancing safety management in construction workplaces. The dataset includes seven classes of basic worker actions. These actions can be utilized to train a deep learning model to detect unsafe behaviors such as distraction, improper use of personal protective equipment, and safety performance degradation. For instance, the "Talk" and "Smoke" actions can be associated with worker distraction and potential safety hazards, while the "Lack of Safety Equipment" action can indicate improper use of personal protective equipment.

Multiple phases of training are involved in the training of the entire system. For both low- and high-resolution inputs, we train both of them with FC classifiers. An SGD with momentum of 0.9 and learning rate of 0.003 is used with a weight decay of 0.0001 at epochs 30, 60, and 90 to train the models with 110 epochs as well as the cross-entropy loss [107]. For other models, feature extraction module weights are frozen and used. In order to train the imaginary, the loss of belief ℓ_{belief} from Eq. 11 is used in conjunction with the teacher forcing routine described in [104]. The decay factor of F_r in our experiment is 0.96, and the warm-up period $\left(t_{\text{warm}}\right)$ is five frames. The module of spatial sampler with the three-head FC is trained utilizing $\ell_{\text{class}} = \sum_{h=1}^{3} \theta_h \ell_h$, where ℓ_h is the cross-entropy as well as scaled by the corresponding factor θ_h. The training process involves the end-to-end training of the temporal sampler alongside the fixed spatial sampler as well as the pre-trained three-head classifier. This training procedure utilizes the total loss $\theta_e \ell_r + \ell_{\text{class}}$, in which ℓ_r represents the reliability loss as defined in Eq. 13, scaled by the corresponding factor θ_e. Our sampling models are trained with Adam optimizer [108] for 50 epochs. This phase of feature extraction focuses on

extracting spatial features only instead of temporal ones. Therefore, we sample only three frames for feature extraction modules during this phase. For better comparison with other frameworks, we use a total of 10 frames for sampling.

4. Preliminary Results and Performance Analysis

4.1. Object Detection Performance

In the context of object detection on the SODA dataset, we observe a range of state-of-the-art methods. Notably, our novel method denoted as "Ours" in Table 1, stands out with a remarkable mAP of 85.27%, signifying superior object detection accuracy. What's equally impressive is the model's efficiency, requiring only 61.98 million parameters, making it an attractive choice for real-world applications with computational constraints. Comparing this against established YOLO variants, YOLOv5 delivers strong results with an mAP of 67.04% and a relatively compact model at 7.01 million parameters. In contrast, YOLOv3 lags behind with an mAP of 57.43%. Notably, "Customize YOLOv4" demonstrates an impressive mAP of 81.47%, though it lacks information about the number of parameters, limiting its broader applicability. Overall, our method exhibits a compelling balance of high accuracy and model efficiency, positioning it as a top contender for object detection on the challenging SODA dataset. However, the choice of the best model should consider specific application requirements and computational resources.

Our study is based on SODA[26] and attempts to detect objects in construction site images in the following ways:

1. Testing the efficacy and efficiency of various feature extraction networks using our proposed approaches;
2. Checking the efficiency of the two proposed attention using base OP and MP-OP;
3. Comparing our proposed methods with different attention structures;
4. Improving the detection of construction objects using RPN input;
5. Showing that different scales have mismatched error rates;
6. demonstrating some visual results.

4.1.1. Different Feature Extraction Networks

In Table 2, the experimental results on SODA's test set, comprising ResNet-50, ResNet-101, and ResNet-152, show different backbone network results. Using the combination of our module and $GFLOPs/FPS$, #Params and mAP, we compare improvements in $GFLOPs/FPS$, #Params and mAP. Our attentions are combined with the backbone, so we observe a 3.95, 4.7, and 3.57 percent increase in mAP for ResNet-50, ResNet-101, and ResNet-152, respectively. Furthermore, model efficiency is compared based on #Params and $GFLOPs/FPS$. It involves around 155 GFLOPs increment, with an average of 1.80M model parameters, and a reduction in performance of around 5-10 FPS.

Table 1
Comparison with state-of-the-art methods on SODA [26].

Methods	mAP(%)	#Params (M)
YOLOv3 [109]	57.43	8.67
YOLOv5 [109]	67.04	7.01
YOLOv7-tiny [110]	55.26	6.01
Scaled-YoLOv4 [111]	67.21	8.06
YOLOX [112]	66.48	8.94
YOLOR [113]	65.77	52.50
SOC-YOLO [114]	68.00	7.20
YOLOv7 [110]	65.77	36.48
Customize YOLOv3 [26]	71.22	-
Customize YOLOv4 [26]	81.47	-
Ours	**85.27**	61.98

Our experiments are based on ResNet-101 due to its mAP and computational complexity.

4.1.2. Proposed Units

According to Table 3, we have calculated the combined performance of our proposed units on ResNet-101. The bounding box mAP improves 0.93% and 2.55% when OP base and MP-OP are used. To illustrate the trend of performance change, Fig. 11 shows the mAP radar chart for each category. Our proposed OP-Net model can increase mAP by as much as 4.7% when base OP and MP-OP are combined (i.e., our proposed OP-Net), with some categories showing very significant improvements (BO 7.13%, RE 6.31%, and SL 5.87%). By using base OP, and MP-OP, we were able to further improve feature presentation capabilities. According to the model efficiency, the base OP brings 0.59M model parameters with 51.53 GFLOPs, while the MP-OP brings 0.61M with 51.89 GFLOPs. This combination produces an increment of 154.95 GFLOPs and 1.79M model parameters. As a result of our proposed OP, GFLOPs increase from 289.25 to 444.22, after calculating the similarity matrix to features. According to Table 4, when adding the multi-head structure in our OP module, #Params reduce 0.36M and mAP increments by 0.65%.

4.1.3. Improving RPN Input for Construction Object Detection

A significant benefit of OP-Net is its ability to address complex background problems and low image quality. Construction images have more complex backgrounds due to overhead shots from different angles that show geological structures, different-sized objects, and different object categories. It is detrimental to learning object features in construction object detection when the imaging quality is poor. This directly affects the training of the modeling algorithms. As a result, we reorganize pyramid OP by implementing base OP on pyramid features. Depending on the maximum response layer, pyramid features generate proposals smaller or larger than those in the region proposal network (RPN)[103]. Because of this, the ROI module will be more difficult to train the detection box when the object proposals are

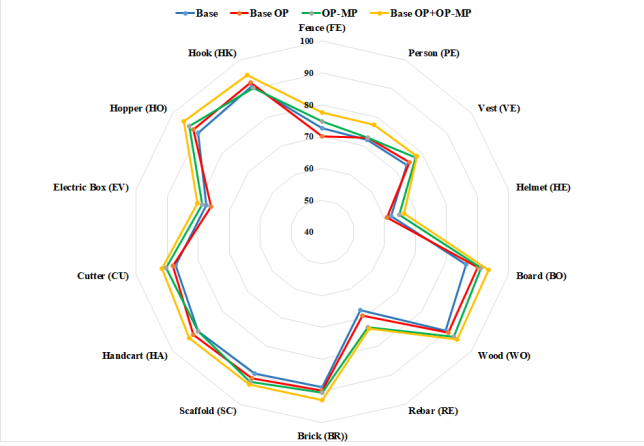

Figure 11: Radar chart for each category of object in SODA [26] dataset. Different detectors are represented by different colored lines. This figure represents the mAP value.

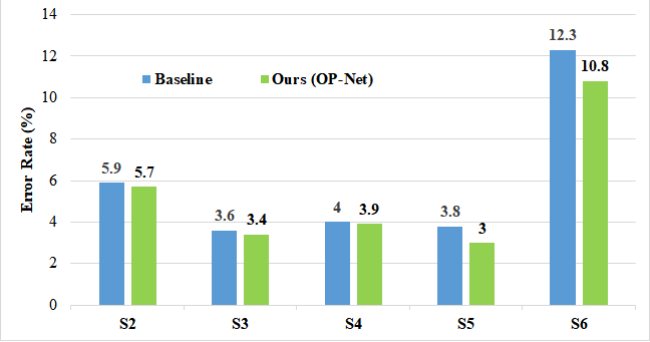

Figure 12: A comparison of the error rate for the baseline method of SODA [26] and the proposed method. The lower the better.

accurate or not. By using OP-Net, the model is able to learn more detailed relation information both between layers and within layers of pyramids. The OP operation should be performed for pyramid features before adding them to the region proposal network.

4.1.4. Error Rates

As a means of demonstrating the impact of the proposed method on each feature level, we define mismatching error rates at different scales within the feature pyramid, that is, objects chosen at different levels are not always consistent with the ground truth. According to Fig. 12, deploying our proposed method (i.e., combining base and MP-OP) resulted in a reduction of mismatching error rates for the layers in the feature pyramid. It is evident that the error rates for high-level features are lower than those for low-level features that are meant for small objects. In levels S2 to S6 there is a 0.2% reduction, 0.2% reduction, 0.1% reduction, 0.2% reduction, and 1.5% reduction in error rate. As a result, our method's effectiveness can be further confirmed.

Table 2
A comparison of the effectiveness of the proposed methods with different networks for extracting features on SODA [26].

Backbone	+Ours	GFLOPs / FPS	#Params (M)	mAP(%)
ResNet-50	✗	212.30/25.1	41.20	79.91
	✓	365.30/14.7	42.98	83.86
ResNet-101	✗	290.31/21.0	60.21	80.57
	✗	**443.31/13.3**	**61.98**	**85.27**
ResNet-152	✗	366.33/17.2	75.88	80.74
	✓	523.19/12.3	77.65	84.31

Note: "+ Ours" denotes the architecture of our proposed base OP and MP-OP attention on the backbone networks.

Table 3
A comparison of different attentions on the SODA [26] test set and ResNet-101 [101] is the backbone.

Base	✓	✓	✓	✓
Base OP	✗	✓	✗	✓
OP-MP	✗	✗	✓	✓
SL	70.36	72.81	74	76.23
FE	72.61	70.14	74.67	77.54
HK	91.01	92.11	90.3	94.73
HO	90.1	91.77	93.49	95.64
EV	77.44	75.82	78.87	80.31
CU	87.29	88.14	90.35	91.64
HA	90.06	91.8	90.12	93.48
SC	89.32	91.12	92.37	93.16
BR	88.71	89.97	90.57	92.95
RE	67.39	69.13	73.12	73.7
WO	89.59	90.71	92.68	94.27
BO	86.37	90.05	91.19	93.5
HE	62.18	60.86	64.8	66.34
VE	73.87	75.02	77.5	78.12
PE	72.19	72.98	72.81	77.51
mAP(%)	80.57	81.5	83.12	85.27
#Params	60.21	60.78	60.8	61.98
GFLOPs/FPS	289.25/20.8	340.8/19.1	341.14/17.1	444.22/13.4

Table 4
A comparison of the effectiveness of our proposed multi-head methods on SODA [26].

Baseline	OP	Multi-head	GFLOPs / FPS	#Params (M)	mAP(%)
✓	✓	✗	461.49/11.9	62.34	84.62
✓	✓	✓	**443.31/13.3**	61.98	**85.27**

Note: "OP" denotes our proposed OP attention on the backbone. "Multi-head" indicates combine multi-head structure in OP blocks.

4.2. Detecting Unsafe Behavior

In the context of developing a model for detecting unsafe worker behavior in construction sites through video analysis, this comparison of state-of-the-art methods on the CMA dataset serves as a valuable reference for understanding the trade-offs between computational efficiency and accuracy. The presented Table 5 illustrates a range of models with varying characteristics, including model size, FLOPS, mAP, and an efficiency trade-off computed as FLOPS over mAP Comparing models with different backbones is also possible with this metric (lower means better).

Table 5
Comparison with state-of-the-art methods on CMA [27].

Model	Size	FLOPS	mAP(%)	Trade-off
CNN+LSTM [115]	224	22.08	36.4	0.606
AVSlowFast [73]	224	39.13	24.2	1.616
TBN [116]	224	150.95	74.83	0.092
TSN [117]	224	33.0	76.1	0.433
TSM [118]	224	32.9	83.0	0.396
Slowfast [49]	224	36.6	78.1	0.468
R3D [119]	224	109.8	70.5	1.557
R(2+1)D [120]	224	117.2	81.9	1.431
TEA [121]	224	35.0	80.6	0.434
ViT-B [**?**]	224	134.8	78.3	1.721
ViT-L [**?**]	224	477.2	81.9	5.826
Swin-B [122]	224	121.0	84.7	1.428
Swin-L [122]	224	272.1	87.4	3.113
MViT-B [123]	224	199.5	81.6	2.444
TimeSformer [124]	224	178.6	82.9	2.154
STR-Transformer [27]	224	202.3	88.7	2.280
SAN19-base	224	160.64	90.72	1.77
Ours	112	44.62	**93.86**	0.475

Among the models, the proposed model stands out as particularly noteworthy. Despite having a compact size of 112, it achieves an impressive mAP of 93.86, making it highly accurate in identifying unsafe behaviors on construction sites. What sets this model apart is its remarkable efficiency trade-off of 0.475, signifying a minimal requirement of computational resources (FLOPS) to achieve high accuracy. This efficiency is particularly advantageous in real-world applications, where computational resources may be limited or cost-effectiveness is a key concern. In contrast, models like "TBN" and "TEA" offer impressive accuracy with minimal computational requirements. "TBN" achieves an mAP of 74.83 with just 150.95 FLOPS, while "TEA" attains an mAP of 80.6 with 35.0 FLOPS. These models exhibit the most favorable trade-offs in terms of computational efficiency.

These findings are of critical importance in the context of your research, where practical deployment of a model in construction sites demands a consideration of computational constraints. The proposed model's superior efficiency trade-off is a compelling feature for real-world applications, as it ensures the timely and cost-effective identification of unsafe behaviors.

4.2.1. Quantitative Analysis

The comparison begins by examining spatial sampling, denoted by models $\mathcal{S}_k$ with varying numbers of extracted ROIs. The baseline "SAN19-base" model, which does not employ spatial sampling, achieves an mAP of 90.72 with a FLOPS requirement of 160.64. As spatial sampling is introduced with $\mathcal{S}_k$, we observe variations in FLOPS, reflecting model complexity. An increase in the number of extracted ROIs, from $\mathcal{S}_0$ to $\mathcal{S}_3$, results in a corresponding rise in computational complexity.

Table 6
A comparison of baselines and spatial sampler $\mathcal{S}$ results. In order to determine the complexity of the model, we include the average number of floating-point operations (FLOPS). Using the Trade-off, we show the amount of FLOPS needed to achieve each accuracy level, based on the average accuracy percentage for each action. Our baseline SAN19 was retrained with TBN [116] using different input modalities. A spatial sampler model is symbolized by $\mathcal{S}_k$, while the number of extracted ROIs is represented by k. Spatial sampling is not performed when $\mathcal{S}_0$ is selected. With $\mathcal{S}_3$, spatial samples are more accurate than baselines and have less complexity.

Model	Backbone	Size	FLOPS	mAP(%)	Trade-off
SAN19-base	Res 50	224	160.64	90.72	1.771
$\mathcal{S}_0$	SAN19	112	138.24	90.56	1.527
$\mathcal{S}_1$	SAN19	112	140.16	**91.23**	1.725
$\mathcal{S}_2$	SAN19	112	142.48	90.04	1.582
$\mathcal{S}_3$	SAN19	112	145.80	88.77	1.642

The key insight from this analysis is the trade-off between computational complexity and model accuracy. As spatial sampling becomes more granular, there is a noticeable trade-off in terms of computational resources and accuracy. For instance, $\mathcal{S}_1$ achieves the highest mAP at 91.23 with a relatively modest increase in FLOPS compared to the baseline, striking a good balance. In contrast, $\mathcal{S}_3$ achieves a lower accuracy (88.77) but demands significantly more FLOPS, underscoring the importance of selecting the appropriate level of spatial granularity for the given application.

The second phase of the comparison explores the interplay between spatial sampling ($\mathcal{S}_0$) and various temporal sampling configurations ($\mathcal{T}_M$). Notably, spatial sampling ($\mathcal{S}_0$) exhibits an mAP of 90.56 with 138.24 FLOPS. Introducing temporal sampling ($\mathcal{T}_M$) causes a tolerable loss of accuracy compared to the scenario without spatial sampling ($\mathcal{S}_0$).

As the number of frames that the temporal sampler is allowed to ignore increases (from $\mathcal{T}_0$ to $\mathcal{T}_4$), the overall computational complexity decreases significantly. This reduction in computational complexity comes with an associated trade-off in model performance. However, $\mathcal{T}_3$, combined with $\mathcal{S}_1$, emerges as an optimal choice, achieving an mAP of 93.86 with a relatively low FLOPS requirement, resulting in an impressive efficiency trade-off of 0.475. This emphasizes the importance of carefully balancing spatial and temporal sampling strategies.

4.2.2. Qualitative Analysis

Beyond quantitative metrics, qualitative aspects play a crucial role. Models with efficient trade-offs, such as $\mathcal{S}_1$ combined with $\mathcal{T}_3$, exhibit a high degree of accuracy in detecting unsafe behaviors while significantly reducing computational demands. These models exhibit a favorable trade-off between model performance and computational efficiency.

Table 7
A comparison between the spatial and temporal sampling on CMA using spatial sampler δ and temporal sampler $\mathcal{J}$. M is the number of frames that the temporal sampler is allowed to ignore in a block, when $\mathcal{T}_M$ is the temporal sampler. Skipped frame percentage (%), pre-scan percentage (%), and fully processed percentage (%) are listed in the table, as well as the FLOPs, as well as the average computational saving of its Spatial Sampler counterpart. Only the first row in this table has spatial sampling, which is copied from the other Table 6. There is a tolerable loss of accuracy when using temporal samplers compared to when spatial sampling is not performed $(\mathcal{S}_0)$.

Model	Full (%)	Skip (%)	Prescan (%)	FLOPs	mAP(%)	Trade-off	Speed up (x)
$\mathcal{S}_0$	100.00	00.00	0.00	138.24	90.56	1.527	-
$\mathcal{T}_1, \mathcal{S}_0$	58.03	00.00	41.97	85.99	92.81	0.927	1.59
$\mathcal{T}_1, \mathcal{S}_1$	50.82	00.00	49.18	82.07	92.98	0.883	1.70
$\mathcal{T}_1, \mathcal{S}_2$	50.03	00.00	49.97	84.06	93.52	0.899	1.77
$\mathcal{T}_1, \mathcal{S}_3$	50.00	00.00	50.00	88.17	93.06	0.947	1.78
$\mathcal{T}_2, \mathcal{T}_0$	33.59	14.06	52.35	54.02	92.52	0.584	2.59
$\mathcal{T}_2, \mathcal{T}_1$	34.05	14.36	51.59	57.01	91.94	0.620	2.45
$\mathcal{T}_2, \mathcal{T}_2$	35.06	12.75	52.19	60.99	92.23	0.661	2.43
$\mathcal{T}_2, \mathcal{T}_3$	32.26	15.03	52.71	59.27	91.23	0.650	2.63
$\mathcal{T}_3, \mathcal{S}_0$	25.63	26.00	48.37	42.09	91.64	0.459	3.30
$\mathcal{T}_3, \mathcal{S}_1$	25.97	25.47	48.55	44.62	**93.86**	0.475	3.12
$\mathcal{T}_3, \mathcal{S}_2$	25.62	25.76	48.61	45.94	93.23	0.493	3.22
$\mathcal{T}_3, \mathcal{S}_3$	25.87	25.19	48.93	48.41	92.73	0.522	3.22
$\mathcal{T}_4, \mathcal{S}_0$	20.53	34.81	44.66	**35.08**	89.85	**0.390**	4.02
$\mathcal{T}_4, \mathcal{S}_1$	21.76	32.07	46.17	37.96	89.89	0.422	3.65
$\mathcal{T}_4, \mathcal{S}_2$	24.42	35.54	40.04	42.95	90.35	0.475	3.44
$\mathcal{T}_4, \mathcal{S}_3$	20.83	34.64	44.53	38.98	88.56	0.440	**3.93**

In terms of speedup, the comparison demonstrates that models with efficient trade-offs lead to substantial improvements in computational efficiency. For instance, $\mathcal{S}_1$ combined with $\mathcal{T}_3$ achieves a speedup of 3.12, indicating that it processes video data more than three times faster than the baseline model ($\mathcal{S}_0$). This speedup is particularly significant for real-time applications and resource-efficient implementations.

Overall, this detailed comparison highlights the intricate relationship between spatial and temporal sampling strategies and their influence on model complexity, accuracy, and computational efficiency. Optimal sampling strategy selection can significantly enhance model efficiency without compromising the ability to detect unsafe worker behavior in construction sites. Researchers and practitioners can leverage these insights to make informed decisions about the most suitable sampling strategies for their specific applications, keeping in mind the critical trade-offs between accuracy and computational resources.

5. Conclusions

The detection of construction objects is complicated due to a complex background and poor image quality. Spacetime feature adjustments are typically approached with elaborate attention mechanisms that are arduous in their computational complexity. For enhanced channel relation, we proposed OP attention that could determine adjust weights by channel. Through extensive experiments on construction image object detection, we implemented OP on a feature pyramid network that is the backbone of a standard object detection network. A small computational overhead is required for the proposed OP-Net to achieve state-of-the-art performance on challenging benchmarks. mAP radar charts displayed robust trends for object detection in each category. As we explore OP-Net's application to more natural scenes, we will explore applying it to different types of subjects. As well as semantic segmentation, object re-identification, and other visual tasks, OP-Net is being explored in other directions. Also, to efficiently recognize unsafe actions in videos from construction sites, this paper proposes a spatial and temporal sampling strategy based on attention that adaptively samples the videos. A high-resolution object of the inputs frames and a low-resolution global inputs frames are provided by the spatial sampler. By evaluating the present attention with the previous imagination, the temporal sampler searches and determines the sampling technique. This study confirms that our proposed method for sampling procedure on top of a different model supporting a self-attention mechanism is feasible. Simpler methods with alternative models can result in competitive complexity as well as performance. Future work should explore real-time, adaptive approaches that leverage temporal-spatial sampling techniques and relational attention models to detect unsafe behaviors. Extend the proposed spatial and temporal sampling strategy to real-time monitoring of construction sites. Implementing a real-time system can provide immediate alerts and interventions for unsafe actions, thus enhancing on-site safety.

6. Disclosures

The authors declare no conflict of interest.

References

[1] D. D. Gransberg, C. M. Popescu, R. Ryan, Construction equipment management for engineers, estimators, and owners, CRC Press, 2006. doi:10.1201/9781420013993.

[2] J. Liu, H. Luo, H. Liu, Deep learning-based data analytics for safety in construction, Automation in Construction 140 (2022) p. 104302. doi:10.1016/j.autcon.2022.104302.

[3] N. D. Nath, A. H. Behzadan, S. G. Paal, Deep learning for site safety: Real-time detection of personal protective equipment, Automation in Construction 112 (2020) p. 103085. doi:10.1016/j.autcon.2020.103085.

[4] I. Awolusi, E. Marks, M. Hallowell, Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices, Automation in Construction 85 (2018) pp. 96–106. doi:10.1016/j.autcon.2017.10.010.

[5] Z. Jiang, D. Fang, M. Zhang, Understanding the causation of construction workers' unsafe behaviors based on system dynamics modeling, Journal of Management in Engineering 31 (6) (2015) p. 04014099. doi:10.1061/(ASCE)ME.1943-5479.0000350.

[6] S. Wu, L. Hou, G. K. Zhang, H. Chen, Real-time mixed reality-based visual warning for construction workforce safety, Automation in Construction 139 (2022) p. 104252. doi:10.1016/j.autcon.2022.104252.

[7] R. A. Haslam, S. A. Hide, A. G. Gibb, D. E. Gyi, T. Pavitt, S. Atkinson, A. R. Duff, Contributing factors in construction accidents, Applied Ergonomics 36 (4) (2005) pp. 401–415. doi:10.1016/j.apergo.2004.12.002.

[8] A. Suraji, A. R. Duff, S. J. Peckitt, Development of causal model of construction accident causation, Journal of Construction Engineering and Management 127 (4) (2001) pp. 337–344. doi:10.1061/(ASCE)0733-9364(2001)127:4(337).

[9] H. J. Müller, J. Krummenacher, Visual search and selective attention, Visual Cognition 14 (4-8) (2006) pp. 389–410. doi:10.1080/13506280500527676.

[10] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) pp. 436–444. doi:10.1038/nature14539.

[11] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T. M. Rose, W. An, Detecting non-hardhat-use by a deep learning method from far-field surveillance videos, Automation in Construction 85 (2018) pp. 1–9. doi:10.1016/j.autcon.2017.09.018.

[12] W. Fang, L. Ding, P. E. Love, H. Luo, H. Li, F. Pena-Mora, B. Zhong, C. Zhou, Computer vision applications in construction safety assurance, Automation in Construction 110 (2020) p. 103013. doi:10.1016/j.autcon.2019.103013.

[13] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, J. C. Niebles, What makes a video a video: Analyzing temporal information in video understanding models and datasets, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 7366–7375. doi:10.1109/cvpr.2018.00769.

[14] Q. Ming, Z. Zhou, L. Miao, H. Zhang, L. Li, Dynamic anchor learning for arbitrary-oriented object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, Association for the Advancement of Artificial Intelligence, 2021, pp. 2355–2363. doi:10.1609/aaai.v35i3.16336.

[15] X. Yang, L. Hou, Y. Zhou, W. Wang, J. Yan, Dense label encoding for boundary discontinuity free rotation detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2021, pp. 15819–15829. doi:10.1109/cvpr46437.2021.01556.

[16] X. Yang, J. Yan, Arbitrary-oriented object detection with circular smooth label, in: European Conference on Computer Vision, Vol. 12353, Springer, 2020, pp. 677–694. doi:10.1007/978-3-030-58598-3_40.

[17] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 779–788. doi:10.1109/cvpr.2016.91.

[18] J. Ding, N. Xue, Y. Long, G.-S. Xia, Q. Lu, Learning roi transformer for oriented object detection in aerial images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2019, pp. 2849–2858. doi:10.1109/cvpr.2019.00296.

[19] J. M. Haut, R. Fernandez-Beltran, M. E. Paoletti, J. Plaza, A. Plaza, Remote sensing image superresolution using deep residual channel attention, IEEE Transactions on Geoscience and Remote Sensing 57 (11) (2019) pp. 9277–9289. doi:10.1109/tgrs.2019.2924818.

[20] C. Wang, X. Bai, S. Wang, J. Zhou, P. Ren, Multiscale visual attention networks for object detection in vhr remote sensing images, IEEE Geoscience and Remote Sensing Letters 16 (2) (2018) pp. 310–314. doi:10.1109/lgrs.2018.2872355.

[21] Q. Wang, S. Liu, J. Chanussot, X. Li, Scene classification with recurrent attention of vhr remote sensing images, IEEE Transactions on Geoscience and Remote Sensing 57 (2) (2018) pp. 1155–1167. doi:10.1109/tgrs.2018.2864987.

[22] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, K. Fu, Scrdet: Towards more robust detection for small, cluttered and rotated objects, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 8232–8241. doi:10.1109/iccv.2019.00832.

[23] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, Q. Sun, Causal intervention for weakly-supervised semantic segmentation, in: Advances in Neural Information Processing Systems, Vol. 33, The MIT Press, 2020, pp. 655–666. doi:10.48550/arXiv.2009.12547.

[24] J. Chen, L. Wan, J. Zhu, G. Xu, M. Deng, Multi-scale spatial and channel-wise attention for improving object detection in remote sensing imagery, IEEE Geoscience and Remote Sensing Letters 17 (4) (2019) pp. 681–685. doi:10.1109/lgrs.2019.2930462.

[25] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 5659–5667. doi:10.1109/cvpr.2017.667.

[26] R. Duan, H. Deng, M. Tian, Y. Deng, J. Lin, Soda: A large-scale open site object detection dataset for deep learning in construction, Automation in Construction 142 (2022) p. 104499. doi:10.1016/j.autcon.2022.104499.

[27] M. Yang, C. Wu, Y. Guo, R. Jiang, F. Zhou, J. Zhang, Z. Yang, Transformer-based deep learning model and video dataset for unsafe action identification in construction projects, Automation in Construction 146 (2023) p. 104703. doi:10.1016/j.autcon.2022.104703.

[28] H. Son, H. Choi, H. Seong, C. Kim, Detection of construction workers under varying poses and changing background in image sequences via very deep residual networks, Automation in Construction 99 (2019) pp. 27–38. doi:10.1016/j.autcon.2018.11.033.

[29] X. Luo, H. Li, D. Cao, F. Dai, J. Seo, S. Lee, Recognizing diverse construction activities in site images via relevance networks of construction-related objects detected by convolutional neural networks, Journal of Computing in Civil Engineering 32 (3) (2018) p. 04018012. doi:10.1061/(ASCE)CP.1943-5487.0000756.

[30] K. Lee, S. Han, Convolutional neural network modeling strategy for fall-related motion recognition using acceleration features of a scaffolding structure, Automation in Construction 130 (2021) p. 103857. doi:10.1016/j.autcon.2021.103857.

[31] S. S. Bangaru, C. Wang, S. A. Busam, F. Aghazadeh, Ann-based automated scaffold builder activity recognition through wearable emg and imu sensors, Automation in Construction 126 (2021) p. 103653. doi:10.1016/j.autcon.2021.103653.

[32] M. F. Antwi-Afari, Y. Qarout, R. Herzallah, S. Anwer, W. Umer, Y. Zhang, P. Manu, Deep learning-based networks for automated recognition and classification of awkward working postures in construction using wearable insole sensor data, Automation in Construction 136 (2022) p. 104181. doi:10.1016/j.autcon.2022.104181.

[33] M. Jung, S. Chi, Human activity classification based on sound recognition and residual convolutional neural network, Automation in Construction 114 (2020) p. 103177. doi:10.1016/j.autcon.2020.103177.

[34] Y.-C. Lee, M. Shariatfar, A. Rashidi, H. W. Lee, Evidence-driven sound detection for prenotification and identification of construction safety hazards and accidents, Automation in Construction 113 (2020) p. 103127. doi:10.1016/j.autcon.2020.103127.

[35] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, C. Li, Computer vision aided inspection on falling prevention measures for steeplejacks in an aerial environment, Automation in Construction 93 (2018) pp. 148–164. doi:10.1016/j.autcon.2018.05.022.

[36] W. Fang, L. Ding, H. Luo, P. E. Love, Falls from heights: A computer vision-based approach for safety harness detection, Automation in Construction 91 (2018) pp. 53–61. doi:10.1016/j.autcon.2018.02.018.

[37] N. D. Nath, A. H. Behzadan, S. G. Paal, Deep learning for site safety: Real-time detection of personal protective equipment, Automation in Construction 112 (2020) p. 103085. doi:10.1016/j.autcon.2020.103085.

[38] R. Xiong, P. Tang, Pose guided anchoring for detecting proper use of personal protective equipment, Automation in Construction 130 (2021) p. 103828. doi:10.1016/j.autcon.2021.103828.

[39] M. Yang, Z. Yang, Y. Guo, S. Su, Z. Fan, A novel yolo based safety helmet detection in intelligent construction platform, in: Intelligent Equipment, Robots, and Vehicles, Springer, 2021, pp. 268–275. doi:10.1007/978-981-16-7213-2_26.

[40] E. Chian, W. Fang, Y. M. Goh, J. Tian, Computer vision approaches for detecting missing barricades, Automation in Construction 131 (2021) p. 103862. doi:10.1016/j.autcon.2021.103862.

[41] W. Fang, B. Zhong, N. Zhao, P. E. Love, H. Luo, J. Xue, S. Xu, A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network, Advanced Engineering Informatics 39 (2019) pp. 170–177. doi:10.1016/j.aei.2018.12.005.

[42] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, IEEE, 2016, pp. 1933–1941. doi:10.1109/cvpr.2016.213.

[43] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, Vol. 27, 2014. doi:10.48550/arXiv.1406.2199.

[44] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: European Conference on Computer Vision, Vol. 9912, Springer, 2016, pp. 20–36. doi:10.1007/978-3-319-46484-8_2.

[45] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2015, pp. 4489–4497. doi:10.1109/iccv.2015.510.

[46] D. Tran, J. Ray, Z. Shou, S.-F. Chang, M. Paluri, Convnet architecture search for spatiotemporal feature learning, arXiv preprint (8 2017). doi:10.48550/arXiv.1708.05038.

[47] Q. Vadis, J. Carreira, A. Zisserman, Action recognition? a new model and the kinetics dataset, arXiv preprint (2017). doi:10.48550/arXiv.1705.07750.

[48] C. Feichtenhofer, X3d: Expanding architectures for efficient video recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 203–213. doi:10.1109/cvpr42600.2020.00028.

[49] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 6202–6211. doi:10.1109/iccv.2019.00630.

[50] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 6450–6459. doi:10.1109/cvpr.2018.00675.

[51] B. Jiang, M. Wang, W. Gan, W. Wu, J. Yan, Stm: Spatiotemporal and motion encoding for action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 2000–2009. doi:10.1109/iccv.2019.00209.

[52] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, L. Wang, Tea: Temporal excitation and aggregation for action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 909–918. doi:10.1109/cvpr42600.2020.00099.

[53] J. Lin, C. Gan, S. Han, Tsm: Temporal shift module for efficient video understanding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 7083–7093. doi:10.1109/iccv.2019.00718.

[54] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 7794–7803. doi:10.1109/cvpr.2018.00813.

[55] S. Han, S. Lee, A vision-based motion capture and recognition framework for behavior-based safety management, Automation in Construction 35 (2013) pp. 131–141. doi:10.1016/j.autcon.2013.05.001.

[56] D. Roberts, W. Torres Calderon, S. Tang, M. Golparvar-Fard, Vision-based construction worker activity analysis informed by body posture, Journal of Computing in Civil Engineering 34 (4) (2020) p. 04020017. doi:10.1061/(ASCE)CP.1943-5487.0000898.

[57] Z. Li, D. Li, Action recognition of construction workers under occlusion, Journal of Building Engineering 45 (2022) p. 103352. doi:10.1016/j.jobe.2021.103352.

[58] C. Ding, S. Wen, W. Ding, K. Liu, E. Belyaev, Temporal segment graph convolutional networks for skeleton-based action recognition, Engineering Applications of Artificial Intelligence 110 (2022) p. 104675. doi:10.1016/j.engappai.2022.104675.

[59] S. Subedi, N. Pradhananga, Sensor-based computational approach to preventing back injuries in construction workers, Automation in Construction 131 (2021) p. 103920. doi:10.1016/j.autcon.2021.103920.

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

[61] L. Ding, W. Fang, H. Luo, P. E. Love, B. Zhong, X. Ouyang, A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory, Automation in Construction 86 (2018) pp. 118–124. doi:10.1016/j.autcon.2017.11.002.

[62] X. Luo, H. Li, Y. Yu, C. Zhou, D. Cao, Combining deep features and activity context to improve recognition of activities of workers in groups, Computer-Aided Civil and Infrastructure Engineering 35 (9) (2020) pp. 965–978. doi:10.1111/mice.12538.

[63] J. Yang, Z. Shi, Z. Wu, Vision-based action recognition of construction workers using dense trajectories, Advanced Engineering Informatics 30 (3) (2016) pp. 327–336. doi:10.1016/j.aei.2016.04.009.

[64] S. Jung, J. Jeoung, H. Kang, T. Hong, 3d convolutional neural network-based one-stage model for real-time action detection in video of construction equipment, Computer-Aided Civil and Infrastructure Engineering 37 (1) (2022) pp. 126–142. doi:10.1111/mice.12695.

[65] J. Kim, S. Chi, Action recognition of earthmoving excavators based on sequential pattern analysis of visual features and operation cycles, Automation in Construction 104 (2019) pp. 255–264. doi:10.1016/j.autcon.2019.03.025.

[66] M. Bügler, A. Borrmann, G. Ogunmakin, P. A. Vela, J. Teizer, Fusion of photogrammetry and video analysis for productivity assessment of earthwork processes, Computer-Aided Civil and Infrastructure Engineering 32 (2) (2017) pp. 107–123. doi:10.1111/mice.12235.

[67] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Q. Weinberger, Deep networks with stochastic depth, in: Proceedings of the European Conference on Computer Vision, Vol. 9908, Springer, 2016, pp. 646–661. doi:10.1007/978-3-319-46493-0_39.

[68] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, R. Feris, Blockdrop: Dynamic inference paths in residual networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 8817–8826. doi:10.1109/cvpr.2018.00919.

[69] X. Wang, F. Yu, Z.-Y. Dou, T. Darrell, J. E. Gonzalez, Skipnet: Learning dynamic routing in convolutional networks, in: Proceedings of the European Conference on Computer Vision, Vol. abs/1711.09485, Springer, 2018, pp. 409–424. doi:10.1007/978-3-030-01261-8_25.

[70] Y. Wang, F. Sun, D. Li, A. Yao, Resolution switchable networks for runtime efficient image recognition, in: Proceedings of the European Conference on Computer Vision, Vol. 12360, Springer, 2020, pp. 533–549. doi:10.1007/978-3-030-58555-6_32.

[71] B. Uzkent, S. Ermon, Learning when and where to zoom with deep reinforcement learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 12345–12354. doi:10.1109/cvpr42600.2020.01236.

[72] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, J. C. Niebles, What makes a video a video: Analyzing temporal information in video understanding models and datasets, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 7366–7375. doi:10.1109/cvpr.2018.00769.

[73] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, C. Feichtenhofer, Audiovisual slowfast networks for video recognition, arXiv preprint (2020). doi:10.48550/arXiv.2001.08740.

[74] B. Korbar, D. Tran, L. Torresani, Scsampler: Sampling salient clips from video for efficient action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 6232–6242. doi:10.1109/iccv.2019.00633.

[75] Y. Meng, C.-C. Lin, R. Panda, P. Sattigeri, L. Karlinsky, A. Oliva, K. Saenko, R. Feris, Ar-net: Adaptive frame resolution for efficient action recognition, in: Proceedings of the European Conference on Computer Vision, Vol. 12352, Springer, 2020, pp. 86–104. doi:10.1007/978-3-030-58571-6_6.

[76] Y. Wang, Z. Chen, H. Jiang, S. Song, Y. Han, G. Huang, Adaptive focus for efficient video recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2021, pp. 16249–16258. doi:10.1109/iccv48922.2021.01594.

[77] H. Kim, M. Jain, J.-T. Lee, S. Yun, F. Porikli, Efficient action recognition via dynamic knowledge propagation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2021, pp. 13719–13728. doi:10.1109/iccv48922.2021.01346.

[78] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, in: Advances in Neural Information Processing Systems, Vol. 31, 2018, pp. 9525–9536. doi:10.48550/arXiv.1810.03292.

[79] H. Zhao, J. Jia, V. Koltun, Exploring self-attention for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 10076–10085. doi:10.1109/cvpr42600.2020.01009.

[80] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, C. Schmid, Vivit: A video vision transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2021, pp. 6836–6846. doi:10.1109/iccv48922.2021.00676.

[81] H. Yin, A. Vahdat, J. M. Alvarez, A. Mallya, J. Kautz, P. Molchanov, A-vit: Adaptive tokens for efficient vision transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2022, pp. 10809–10818. doi:10.1109/cvpr52688.2022.01054.

[82] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, C. Feichtenhofer, Mvitv2: Improved multiscale vision transformers for classification and detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2022, pp. 4804–4814. doi:10.1109/cvpr52688.2022.00476.

[83] C.-Y. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, C. Feichtenhofer, Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2022, pp. 13587–13597. doi:10.1109/cvpr52688.2022.01322.

[84] H. Yin, A. Vahdat, J. Alvarez, A. Mallya, J. Kautz, P. Molchanov, Adavit: Adaptive tokens for efficient vision transformer, arXiv preprint (12 2021). doi:10.48550/arXiv.2112.07658.

[85] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229. doi:10.1007/978-3-030-58452-8_13.

[86] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, l. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint 30 (2017) pp. 5998–6008. doi:10.48550/arXiv.1706.03762.

[87] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 10428–10436. doi:10.1109/cvpr42600.2020.01044.

[88] Z. Yang, Z. Dai, R. Salakhutdinov, W. W. Cohen, Breaking the softmax bottleneck: A high-rank rnn language model, in: 6th International Conference on Learning Representations, OpenReview.net, 2017, [Accessed Feb, 26, 2024].
URL https://openreview.net/forum?id=HkwZSG-CZ

[89] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 6848–6856. doi:10.1109/cvpr.2018.00716.

[90] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, OpenReview.net, 2021, [Accessed Feb, 26, 2024]. doi:0.48550/arXiv.2010.11929.
URL https://openreview.net/forum?id=YicbFdNTTy

[91] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019. doi:10.48550/arXiv.1905.09418.

[92] I. Bello, B. Zoph, A. Vaswani, J. Shlens, Q. V. Le, Attention augmented convolutional networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 3286–3295. doi:10.1109/iccv.2019.00338.

[93] Y. Lin, P. Feng, J. Guan, W. Wang, J. Chambers, Ienet: Interacting embranchment one stage anchor free detector for orientation aerial object detection, arXiv preprint (12 2019). doi:10.48550/arXiv.1912.00969.

[94] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 2117–2125. doi:10.1109/cvpr.2017.106.

[95] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2017, pp. 2980–2988. doi:10.1109/iccv.2017.324.

[96] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, Q. Sun, Feature pyramid transformer, in: European Conference on Computer Vision, Springer, 2020, pp. 323–339. doi:10.1007/978-3-030-58604-1_20.

[97] G. Ghiasi, T.-Y. Lin, Q. V. Le, Nas-fpn: Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2019, pp. 7036–7045. doi:10.1109/cvpr.2019.00720.

[98] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: Towards balanced learning for object detection, in: Proceedings of

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2019, pp. 821–830. doi:10.1109/cvpr.2019.00091.

[99] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, J. Sun, Thundernet: Towards real-time generic object detection on mobile devices, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 6718–6727. doi:10.1109/iccv.2019.00682.

[100] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 7132–7141. doi:10.1109/cvpr.2018.00745.

[101] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778. doi:10.1109/cvpr.2016.90.

[102] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255. doi:10.1109/cvpr.2009.5206848.

[103] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, Vol. 28, The MIT Press, 2015, pp. 1137–1149. doi:10.1109/tpami.2016.2577031.

[104] R. J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Computation 1 (2) (1989) pp. 270–280. doi:10.1162/neco.1989.1.2.270.

[105] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, arXiv preprint 28 (2015) pp. 802–810. doi:10.48550/arXiv.1506.04214.

[106] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint (11 2016). doi:10.48550/arXiv.1611.01144.

[107] N. Qian, On the momentum term in gradient descent learning algorithms, Neural Networks 12 (1) (1999) pp. 145–151. doi:10.1016/S0893-6080(98)00116-6.

[108] D. Kingma, Adam: a method for stochastic optimization, in: The International Conference on Learning Representations, 2014. doi:10.48550/arXiv.1412.6980.

[109] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint (4 2018). doi:10.48550/arXiv.1804.02767.

[110] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2023, pp. 7464–7475. doi:10.1109/cvpr52729.2023.00721.

[111] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Scaled-yolov4: Scaling cross stage partial network, in: Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern Recognition, IEEE, 2021, pp. 13029–13038. doi:10.1109/cvpr46437.2021.01283.

[112] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv preprint (7 2021). doi:10.48550/arXiv.2107.08430.

[113] C.-Y. Wang, I.-H. Yeh, H.-Y. M. Liao, You only learn one representation: Unified network for multiple tasks, arXiv preprint 39 (2021) pp. 691–709. doi:10.48550/arXiv.2105.04206.

[114] M. Park, J. Bak, S. Park, Small and overlapping worker detection at construction sites, Automation in Construction 151 (2023) p. 104856. doi:10.1016/j.autcon.2023.104856.

[115] L. Ding, W. Fang, H. Luo, P. E. Love, B. Zhong, X. Ouyang, A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory, Automation in construction 86 (2018) pp. 118–124. doi:10.1016/j.autcon.2017.11.002.

[116] E. Kazakos, A. Nagrani, A. Zisserman, D. Damen, Epic-fusion: Audio-visual temporal binding for egocentric action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 5492–5501. doi:10.1109/iccv.2019.00559.

[117] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: European Conference on Computer Vision, Vol. 9912, Springer, 2016, pp. 20–36. doi:10.1007/978-3-319-46484-8_2.

[118] J. Lin, C. Gan, S. Han, Tsm: Temporal shift module for efficient video understanding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2019, pp. 7083–7093. doi:10.1109/iccv.2019.00718.

[119] D. Tran, J. Ray, Z. Shou, S.-F. Chang, M. Paluri, Convnet architecture search for spatiotemporal feature learning, arXiv preprint (8 2017). doi:10.48550/arXiv.1708.05038.

[120] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 6450–6459. doi:10.1109/cvpr.2018.00675.

[121] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, L. Wang, Tea: Temporal excitation and aggregation for action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 909–918. doi:10.1109/cvpr42600.2020.00099.

[122] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, IEEE, 2021, pp. 10012–10022. doi:10.1109/iccv48922.2021.00986.

[123] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, IEEE, 2021, pp. 6824–6835. doi:10.1109/iccv48922.2021.00675.

[124] G. Bertasius, H. Wang, L. Torresani, Is space-time attention all you need for video understanding?, in: International Conference on Machine Learning, Vol. 2, Proceedings of Machine Learning Research, 2021, p. 4. doi:10.48550/arXiv.2102.05095.