Neural Computing and Applications (2022) 34:4207–4220
https://doi.org/10.1007/s00521-021-06489-3

ORIGINAL ARTICLE

Machine-learning-based top-view safety monitoring of ground workforce on complex industrial sites

Gelayol Golcarenarenji[1] · Ignacio Martinez-Alpiste[1] · Qi Wang[1] · Jose Maria Alcaraz-Calero[1]

Received: 7 March 2021 / Accepted: 30 August 2021 / Published online: 22 October 2021

Abstract
Telescopic cranes are powerful lifting facilities employed in construction, transportation, manufacturing and other industries. Since the ground workforce cannot be aware of their surrounding environment during the current crane operations in busy and complex sites, accidents and even fatalities are not avoidable. Hence, deploying an automatic and accurate top-view human detection solution would make significant improvements to the health and safety of the workforce on such industrial operational sites. The proposed method (CraneNet) is a new machine learning empowered solution to increase the visibility of a crane operator in complex industrial operational environments while addressing the challenges of human detection from top-view on a resource-constrained small-form PC to meet the space constraint in the operator's cabin. CraneNet consists of 4 modified ResBlock-D modules to fulfill the real-time requirements. To increase the accuracy of small humans at high altitudes which is crucial for this use-case, a PAN (Path Aggregation Network) was designed and added to the architecture. This enhances the structure of CraneNet by adding a bottom-up path to spread the low-level information. Furthermore, three output layers were employed in CraneNet to further improve the accuracy of small objects. Spatial Pyramid Pooling (SPP) was integrated at the end of the backbone stage which increases the receptive field of the backbone, thereby increasing the accuracy. The CraneNet has achieved 92.59% of accuracy at 19 FPS on a portable device. The proposed machine learning model has been trained with the Standford Drone Dataset and Visdrone 2019 to further show the efficacy of the smart crane approach. Consequently, the proposed system is able to detect people in complex industrial operational areas from a distance up to 50 meters between the camera and the person. This system is also applicable to the detection of any other objects from an overhead camera.

Keywords Smart telescopic crane · Human detection · Complex industrial sites · Deep learning

1 Introduction

Telescopic cranes are widely employed in construction, oil and gas, maritime ports, transportation and manufacturing industries across the globe. However, the crane operations can be precarious [1, 2] to their surrounding environment due to the operator's lack of visibility to detect workers [3] and can cause accidents and even fatalities on industrial sites [4]. Hence, it is of myriad importance to improve safety by minimising the risks of crane operations. These crane-related hazards can be significantly reduced by automated monitoring and detecting the workforce in real-time from top-view, which allows the operator to take immediate safety actions. Human detection has wide-spread applications in surveillance and monitoring systems, person counting and tracking, search and rescuing missing people [5–7], and action and behavioural understanding, among others [8, 9]. Human detection from top-view has also attracted attention in recent years since it deals better with the occlusion problems and can cover a wider area of the scenes compared with frontal-view based human detection [10]. In addition, overhead human detection can decrease privacy issues as only the detection of human bodies rather than faces is involved. However, top-view human detection is still challenging due to complex backgrounds, small-sized human objects, different illumination, a wide range of pose variations, occlusion, and the

✉ Gelayol Golcarenarenji
G.Golcarenarenji@uws.ac.uk

[1] School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley, United Kingdom

industrial sites being cluttered and dynamically changing [10]. As a result, our objective is to design and develop a new end-to-end real-time machine-learning-based human detection system (CraneNet) being able to detect people in cluttered industrial sites with high accuracy and efficiency considering the challenges involved in top-view human detection and more applicable to industrial use cases and especially crane sites. The application of the developed integrated system has been validated in real crane operation sites, such the one shown in Fig. 1, located at Stirling, Scotland.

Figure 2 also presents a high-level overview of the proposed top-view real-time human detection system for industrial sites.

As illustrated in Fig. 2, the system comprises a video camera subsystem attached to the crane hook to capture the view of the concerned area and send the video stream to a small-form PC (Jetson AGX Xavier in this study) for space-constrained industrial operational environments such as the cabin of a crane or small in situ operating office. This small-form PC hosts the machine learning platform and algorithms to detect any human object entering the proximity by analysing the video input from the camera.

As a result, our contributions are summarised as follows:

- Design and develop a top-view object detection system on compact and power-constrained devices suitable for industrial operational sites.
- Presentation of a new convolutional neural network (CNN) design to detect small-sized human objects from cranes on industrial sites.
- Training and validation of CraneNet and state-of-the-art techniques with the collected dataset and a public dataset to compare the performance.
- Perform both qualitative and quantitative testing evaluation of CraneNet and comparing it with the state-of-the-art algorithms under study using images captured by a camera attached to the crane hook.
- Application of the proposed solution in a real-world use case in a complex industrial scenario.

Fig. 1 Industrial crane operation facility in Stirling, Scotland

Fig. 2 The system for top-view detection of industrial sites. The system includes a video camera subsystem attached to the crane hook to capture and send the video stream to the small-form PC. The small-form PC hosts the machine learning platform and algorithms to detect the human objects

The rest of the paper is organised as follows. Section 2 presents the related work. Section 3 describes the design of the proposed algorithm to detect humans, followed by the experimental setup in Section 4. Section 5 discusses the results of the proposed algorithm. Section 6 concludes the paper.

2 Related work

This section reviews state-of-the-art work related to human detection, with a focus on techniques used in this paper for small object detection.

2.1 Top-view human detection techniques

Various machine learning and deep-learning based techniques for human detection are based on frontal-view imagery. The main methods adopted in top-view human detection are traditional and deep-learning techniques. In traditional methods, features such as hard-hat or high visibility jackets are detected [11, 12] rather than performing directly human detection. Deep learning-based techniques have also been explored focusing more on direct human detection techniques [2, 13]. Two-stage and one-stage object detectors are two major categories of CNN-based methods. Examples of two-stage detectors are fast regions with convolutional neural networks (R-CNN) [14], Faster RCNN [15, 16], and Mask R-CNN [17] in which the classification is performed after the Region of Interest (ROI) is localised. These detectors are accurate but computationally expensive and thus are not suitable for real-

time object detection on low-power devices. One-stage detectors such as YOLO series [18–20], "You Only Look Twice (YOLT)" [21] and "Single Shot Detector (SSD)" [22] are more common. However, their performances are inferior in accuracy. In these detectors, ROIs are localised and classified in one go.

2.2 Small object detection

An object is called small when it is less than 1% of the image area. Small object detection is still one of the most unresolved and challenging detection problems due to the extraction of feature information of small objects being difficult with only a few pixels. It is very hard for standard detectors to distinguish small objects from generic clutter in the background. The features of small objects filter out during the downsampling process in deeper layers of the CNN and thereby never get detected or classified. In addition, feature maps with low resolution and longer distances from the ROI make detection of small objects with various poses and viewpoints even more difficult. Lastly, there is no large publicly available dataset for small objects. While the mean average precision using the state-of-the-art detectors on a dataset like PASCAL VOC is 76.3%, the mAP of a state-of-the-art on a dataset with only small objects is just 27% [23].

2.3 Path aggregation network (PANet)

Inspired by the Feature Pyramid Network (FPN) [24], PAN [25, 26] is a method that improves the accuracy of small object detection by adding an additional bottom-up path augmentation to combine features from initial layers with more detailed information and the deeper layers with more meaningful information as both information is needed to improve small object detection. In other words, the data path between lower layers and top layer is reduced using the PAN architecture. The PAN architecture was modified compared to that of YOLOv4 (2 upsamples) by adding an extra upsample to retain more shallow features due to its importance in small object detection.

2.4 SPP

Spatial Pyramid Pooling (SPP) [27] enhances the receptive field by using relatively large $k \times k$ max-pooling and increases the accuracy of the proposed design. It is also vigorous to object deformation and perform some information aggregation at a deeper stage of the neural network. A modified SPP module in strides was used to enhance the receptive field of the backbone in CraneNet with concatenation of max-pooling outputs with kernel size $k \times k$, where $k = 4, 8, 16$.

2.5 Previous work

This subsection provides an overview of previous work related to top-view human detection. The comparative analysis is summarised in Table 1 to highlight representative published results related to top-view human detection, in comparison with the proposed solution in this paper (highlighted in green).

In one study, Maheshwari et al. [28] have used state-of-the-art Ada-Boost classifier for detection and kalman filter for tracking of people from top-view perspective. Although a high accuracy of 97% was achieved, and the authors mentioned different altitudes, the results are not demonstrated at high altitudes and cluttered environments, which make the detection significantly more complex. In [29], three state-of-the-art models were compared including fully convolutional neural network (FCN) [30] with Resnet-101 architecture, U-Net [31] with encoder-decoder architecture and DeepLabV3 model [32] with encoder-decoder architecture. The experimental results reveal the effectiveness and performance of segmentation models by achieving mean Intersection over Union (mIoU) of 80%, 82% and 84% for FCN, U-Net, and DeepLabv3, respectively. However, these models were applied on images rather than videos. In addition, the aforementioned architecture is not fast enough to be applicable in real-time applications. In [33], a combination of RCNN and GOTURN algorithms was investigated for person detection from the top view. Although very high-accuracy of 90-93% was achieved, it is not applicable to real-time applications due to being a heavy-weight model. In [34], SSD was evaluated with various datasets collected from unreal engine (UE) and real-world from the distance of 20 to 26 meters. The detector achieved the AP of 49.03% and 39.24% for UE dataset and 75.23% using real-world dataset. However, both accuracy and speed are not sufficient for our top-view human detection use case. In [35], YOLOv3 was utilised to detect human from the top view. Although a high accuracy of 92% was achieved, this is a heavy-weight model not suitable for power constrained devices. In another study [36], a computer vision based person counting system was presented based on background subtraction capturing people by an overhead camera installed at about 7 meters height. The proposed algorithm achieved an average accuracy of 98% for person detection and counting. Although they achieved high accuracy, the speed has not been mentioned. In addition, the results are reported at very low altitudes. Similarly, in [37–39], the speed was not mentioned for further evaluation. In [3], Retina-net was used for load-view human detection using a simulated environment. The accuracy and speed of the study were not enough for our use-case. In

Table 1 Comparison of previous works

Ref	Objective	Algorithm	Exec Environment	Platform	Accuracy (%)	Speed (FPS)	Model Size (MB)	Altitude (m)
[28]	Person	Ada-boost/Kalman filter	Raspberry Pi3B	NG	97	Up to 40	NG	NG
[37]	Person	RHOG-SVM	PC	OpenCV	94/96	NG	NG	5
[34]	Person	SSD	PC	Caffe	49.0/72.3	4.6/4.8	NG	20-26
[29]	Person	FCN/U-net	PC	OpenCV	80/82	17/17.5 ms	NG	4
		DeepLabV3			84	18 ms		
[40]	Person	RHOG	PC	OpenCV	95	0.7 ms per bounding box	NG	5
[38]	Person	SVM	PC	LIBSVM	89.5	NG	NG	NG
[36]	Person	Morphological operation	PC	OpenCV	95	NG	NG	7
[35]	Person	YOLOv3	PC	OpenCV	95	NG	NG	NG
[3]	Person	RetinaNet	PC	NG	66.84/53.13	6.6/6.5	NG	25-35
[39]	Person	Mixture of Gaussian	PC	OpenCV	96	NG	NG	7
[33]	Person	Fast-RCNN+ GOTURN	PC	OpenCV	90-94	NG	NG	3
TP	Person	Own approach	Jetson Xavier	Tensorflow	92.59	19	42.9	Up to 50
TP	Person	Own approach	Jetson Xavier	Tensorflow	93.2/94.3/93.4/92.0/90.1	19	42.9	10/20/30/40/50

TP = this paper; NG = not given

[40], a robust algorithm was proposed for detecting people from overhead views. The accuracy of 95% and the average processing time of 0.7 ms per detection window were obtained. However, the result has not been reported at high altitudes which makes the detection more challenging. All of the studies above except [28] have been conducted on powerful PCs and not suitable on embedded systems such as Jetson Xavier.

In summary, none of the above studies has covered or considered the essential factors of human detection [41, 42]. Moreover, none of the previous studies has considered the best trade-off between speed and accuracy suitable for our top-view human detection use case. In addition, CraneNet is able to detect humans up to 50 meters which was not the case in previous studies. Therefore, in this work, a learning-based model is provided to detect person from overhead view at different altitudes up to 50 meters in close real-time with high accuracy on a power constrained device (Jetson Xavier). To achieve this, a novel object detection algorithm was proposed by using four modified ResBlock-D in the backbone to improve the speed, a modified SPP module at the end of the backbone to improve the accuracy and a modified PAN to increase the accuracy of small object detection.

3 Proposed smart crane approach

3.1 Design of object detection algorithm

In the proposed system, the small-form PC receives the video stream from the camera, and it decodes the frames and prepares them to enter the neural network (pre-processing). Once the frame is pre-processed in the correct format, it enters the proposed neural network (CraneNet) embedded on the small-form PC. To improve object detection speed, modified ResBlock-D [43] was proposed to fulfill the real-time requirements. To increase the accuracy of small humans at high altitudes which is crucial for this use-case, a PAN (Path Aggregation Network) module was integrated to the architecture. This enhances the structure of CraneNet by adding a bottom-up path to spread the low-level information. Furthermore, three output layers were also used in CraneNet to further improve the accuracy of small objects. Spatial Pyramid Pooling (SPP) [27] was also added at the end of the backbone stage which is vigorous to object deformation and perform some information "aggregation" at a deeper stage of the neural network. The architecture of CraneNet is illustrated in Fig. 3.

As shown in Fig. 3, the backbone starts with a 3×3 convolution with the number of filter being 64. This was

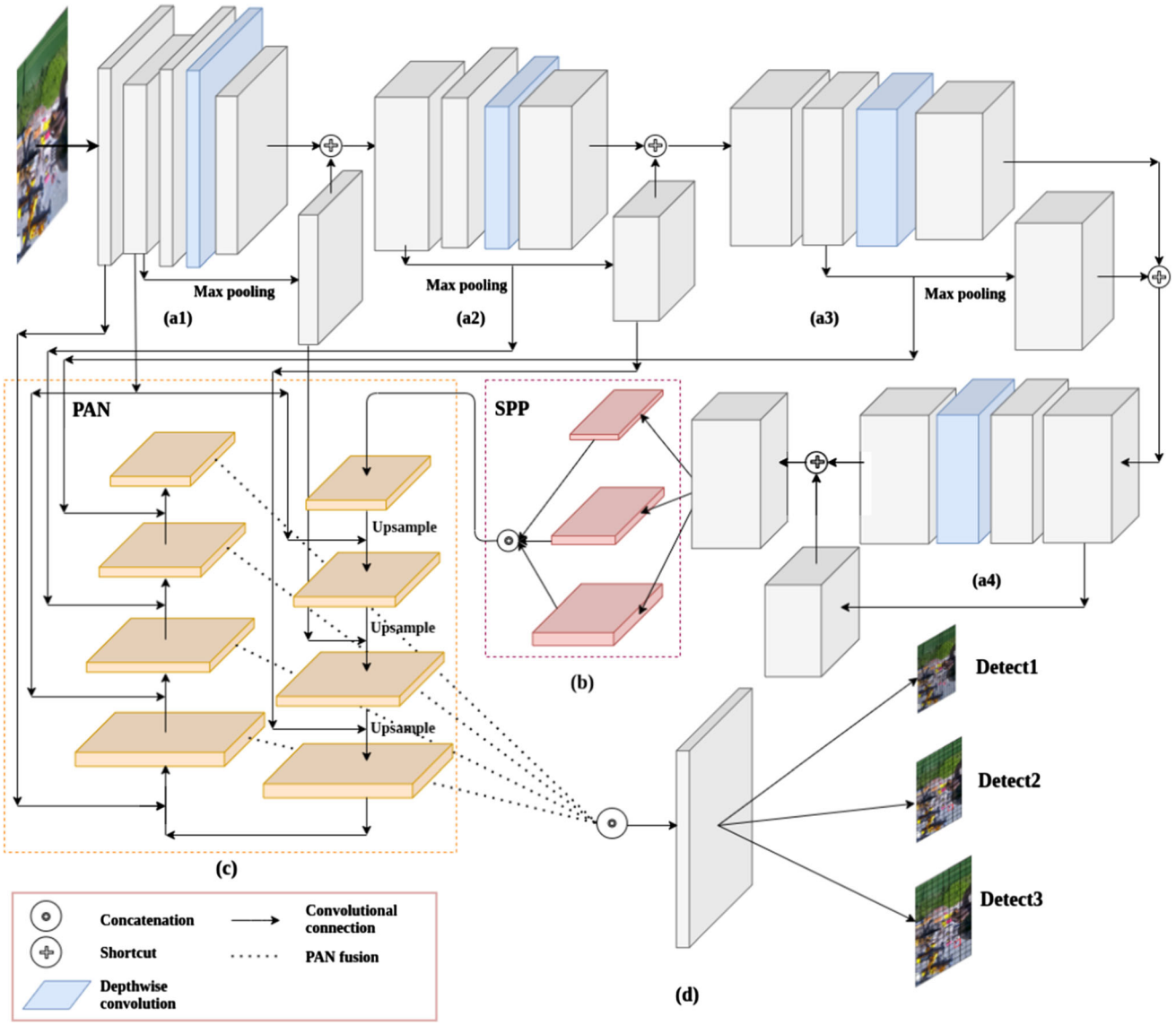


Fig. 3 The architecture of the proposed algorithm. Four modified ResBlock-D like modules were proposed in the backbone (labeled as (a)) including two paths: first path with one 1×1 convolution, one 3×3 depth-wise followed by a 1×1 convolution and the second path with a 2×2 max pooling and a 1×1 convolution. The SPP module (labelled as (b)) includes the max-pooling with strides of 4, 8 and 16. The PAN section (labelled as (c)) consists of with a top-down pathway with three up-samplings. The extra bottom-up path augmentation was gradually down-sampled. The features from the bottom-up path were then concatenated, and 1×1 convolution was run on the result

then accompanied by a maxpooling. Inspired by ResBlock-D like modules in [43], four modified blocks were proposed in the backbone of the proposed neural network (labelled as a1, a2, a3, and a4). The modified module contains two paths. The first path includes one 1×1 convolution, one 3×3 depth-wise and groups of 32 to reduce the number of output parameters and to speed up the proposed model and to replace the traditional convolution in [43] followed by a 1×1 convolution. The second path includes a 2×2 max pooling and a 1×1 convolution. A shortcut connection was used at the end of the second path. The number of filters of the first module were selected 128, 64, 64 and 128. The number of filter were 256, 128, 128 and 256 for the second module, and the number of filters for the third and the forth modules were 512, 256, 256, 512. In addition, the maxpooling of the second path has been eliminated in this module. This was proposed to deal with small object detection.

Secondly, Spatial Pyramid Pooling (SPP) [27] was utilised (labelled as b) at the end of the backbone. However, the max-pooling strides were selected as 4, 8 and 16 different from those of [44]. A PAN (Path Aggregation Network) module [25, 26] was also integrated (labelled as c) in the algorithm which consists of a top-down pathway with

three up-samplings and the number of filters being 128, 256, 128 and 64, respectively. The extra bottom-up path augmentation was gradually down-sampled with factors of 16, 8, 4 and 2, respectively. The number of filters used were 32. The features from the bottom-up path were then concatenated, and 1×1 convolution was run on the result (labelled as d). The YOLOv3 headers were used for the final detections. The detection heads applied the anchors to outputs the coordinates, the class of object being detected and probability estimated for such detection (level of confidence) shown on the screen. The resolution of three scales of the feature map was 19×19, 38×38 and 76×76 for the three detection heads. The combination and integration of modified ResBlock-D like backbone blocks, SPP module and PAN created the CraneNet. To sum up, to improve the accuracy of CraneNet, a modified SPP module was used to enhance the receptive field of the backbone in our approach with a concatenation of max-pooling outputs with kernel size $k \times k$, where $k = 4$, 8, 16 and stride 1. This relatively large $k \times k$ max-pooling increases the receptive field of the backbone and thereby increases the accuracy of the proposed design. It is also vigorous to object deformation and perform some information aggregation at a deeper stage of the neural network. To increase the accuracy of small humans at high altitudes (top-view object detection) which is crucial for this use case, a modified PAN (Path Aggregation Network) module was implemented and added into the architecture. Inspired by the Feature Pyramid Network (FPN), PAN is a method that improves the accuracy of small object detection by adding an additional bottom-up path augmentation to combine features from initial layers with more detailed information and the deeper layers with more meaningful information as both information is needed to improve small object detection. In other words, the data path between lower layers and top layer is reduced by using the PAN architecture. The PAN architecture was modified compared to that of YOLOv4 (2 upsamplings) by adding an extra upsample (3 upsamplings) to retain more shallow features due to its importance in small object detection. The extra bottom-up path augmentation was gradually downsampled (4 downsamplings) compared to the 2 downsamplings in YOLOv4. To improve object detection speed, four modified ResBlock-D were proposed in the backbone. The modified module contains two paths. The first path includes one 1×1 convolution, one 3×3 depth-wise to speed up the Cranenet followed by a 1×1 convolution. The second path includes a 2×2 max pooling and a 1×1 convolution. A shortcut connection was used at the end of the second path. The max-pooling of the second path in the fourth module was eliminated in this module to deal with and improve the small object detection.

3.2 Design of object detection pipeline

The proposed CraneNet should be executed in an optimised machine learning platform. As convolutional neural networks are computationally expensive, the results may produce delays. The achievement of real-time execution mainly depends on two factors: execution environment and the algorithm complexity. In order to mitigate the delays produced by these factors, we designed and implemented a Multi-Thread Pipeline software illustrated in Fig. 4.

The execution platform is divided into two threads: the main thread and the detection thread. The main thread is accountable for receiving and decoding the video frames from the camera on the hook of the crane. It also chooses whether a frame should be processed or discarded by the proposed CNN model. The decision to process a frame depends on the availability of the detection thread. The availability is provided by a blocking queue (BQueue) occupancy of size one. If the queue is already full, it means that the detection thread is currently executing the object detector algorithm, and thereby the queue is blocked and the current frame is discarded. By contrast, if the BQueue is empty, the main thread will store the frame in BQueue. The post-processed frames will then be sent to a secondary blocking queue (SQueue) to be then shown on the screen.

The main task carried out by the detection thread is the execution of the object detector algorithm on frames provided by the main thread. After dequeuing a frame from the BQueue, the detection thread starts a three-stage process to provide the detection results. First, a pre-processing step will prepare the image as an input for the neural network. The detection stage executes the CNN model and provides three results: the class of the object, the location of the object and the confidence score. This step is the most time-consuming task.

4 Experimental setup

4.1 Dataset

A Reolink optical camera was employed to fulfil the aims of the project. The camera was attached to the hook of the crane pointing down to the ground and moved at different heights between 10 and 50 meters in various industrial sites. The live videos were recorded in 2K from top-view, and the images were extracted from the collected videos for further processing. The experiments took place during the daytime in cloudy and sunny weather conditions. Business team ground personnel (men and women) were asked to perform their routine activities in different operational venues. The people were wearing special uniforms, hi-vis

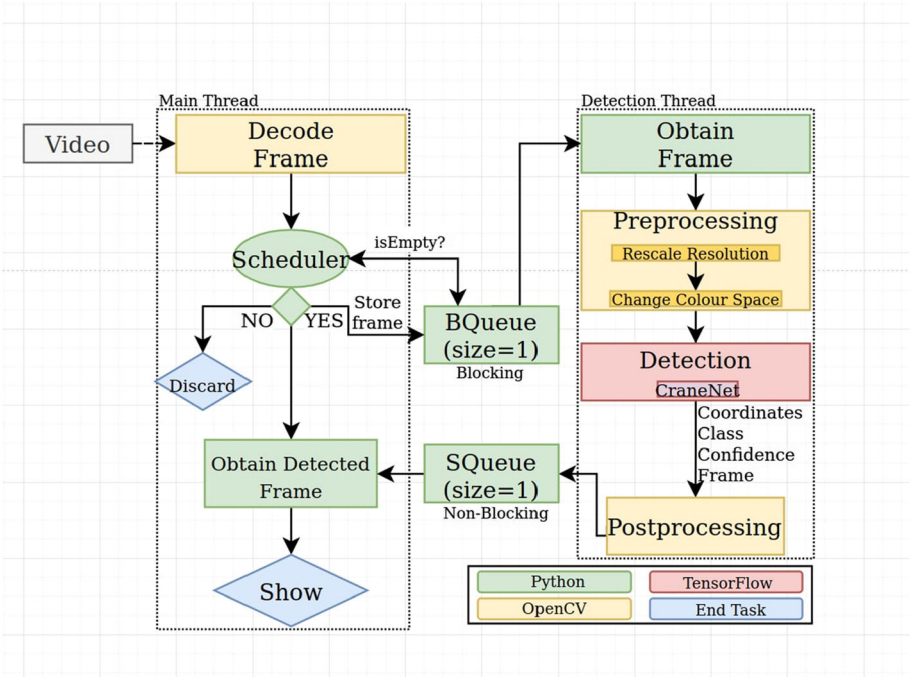


Fig. 4 The Execution Platform. The main thread receives and decodes the video frames from the camera and decides whether a frame should be processed or discarded by the CNN model. The detection thread executes the object detector algorithm on frames provided by the main thread. Opencv was used in components colored in yellow. Components in green have been performed using python. Tensorflow was used for the detection model

jackets and hard hats in red, green, white or yellow. The data were collected at different industrial sites with various backgrounds. More obstacles and distractions (traffic cones, wheelie waste bins,...) were added to the experimental scene to make it more cluttered and challenging for human detection. In addition, to collect the required dataset, various poses, figures, postures, scales, angles, orientations, sizes and altitudes were taken into consideration. Finally, the detection of single and multiple ground personnel was shown on a screen at various altitudes from 10 to 50 meters.

Totally, 20,234 images (10,336 positive samples and 9898 negative samples) were extracted and manually annotated to create the training dataset.

YOLO-Mark2 [45] was used to label the images required for the dataset. To further increase the performance of the developed model and avoid class imbalance, synthetic image was added using copy-paste [46] and data augmentation strategies [47, 48] to generate same number of positive and negative images. Different geometric transformation techniques such as rotation, flipping, blurring, Gaussian noise and various brightness and contrast were used to create synthetic images.

According to Table 2, the average scale of human size is 1802×7176 at 10 m, 209×2376 at 20 m, 135×621 at 30 m, 80×400 at 40 m, and 72×260 at 50 m, respectively.

Table 2 The size of human at different altitudes

Altitude (m)	Min size (px)	Max size (px)
10	1440	7272
20	300	6336
30	165	6120
40	130	5986
50	108	5916

4.2 Hyperparameters

This subsection determined the hyperparameters used to achieve the aims of this study. To determine the best anchor boxes for the dataset, the size of the input images (608 × 608 pixels) and the number of the anchors boxes (9 in this case) were adopted to recalculation the anchor boxes using k-means technique [49]. The same values of the hyperparameters were used for all the algorithms to conduct a true comparison of the algorithms. The total number of training iterations was set to 30,000. The initial learning rate was set to 0.008. A Stochastic Gradient Descent with Warm Restarts (SGDR) [50] was used as the solver. The momentum coefficient was set to 0.9 for the learning policy. A weight decay and the subdivision were set to 0.001 and 8, respectively. The hyperparameters are summarised in Table 3.

The performance of each model was evaluated in terms of two metrics: mAP on validation data and FPS as the speed of the algorithms. The model with best mAP validation was selected for further comparison.

Transfer learning [51] was employed using pre-trained weights obtained from COCO dataset [52] to further increase the performance of the models.

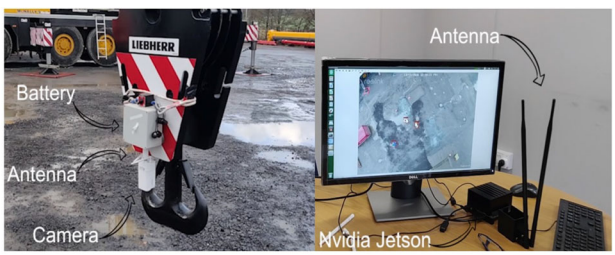
Table 3 Execution hyperparameters

Hyperparameters	Values
Image size in pixel	608 × 608
Number of iteration	30,000
Batch size	64
Initial learning rate	0.008
Solver	SGDR
Momentum coefficient	0.9
Weight decay	0.001

4.3 Evaluation of the approach in a real use case

The proposed object detection system was deployed and evaluated in a real-world scenario in an industrial site located in Stirling (Scotland), with access to a cameras attached to the hook of different cranes.

The video feeds were sent from the customised Reolink camera (Reolink RLC-511W) attached to the crane hook (Fig. 5a) to the proposed CNN-based model embedded on an NVIDIA Jetson Xavier small-form PC connected to a monitor for human detection through the Real-Time Messaging Protocol (RTMP) and Real-Time Streaming protocol (RTSP) (Fig. 5b). The small-form PC is a commercial off-the-shelf platform with 512-core Volta GPU and 64 Tensor Cores. It runs on Linux, with more than 21 (Tera Operations Per Second) TOPS of computation performance and 32GB of RAM. Moreover, it has the ability of working at different power modes. The NVIDIA Jetson Xavier was transformed to an access point using a wireless card which allowed the transmission of the video from the Reolink Camera to the Jetson for human detection. The crane system was elevated to the maximum altitude and then was lowered one step at a time to evaluate and see the result on the screen at various altitudes.



(a) Customised Camera mounted on the crane hook. **(b)** Deployed AI-based system.

Fig. 5 The integration with Reolink camera. The customised camera is attached to the crane hook (**a**). The NVIDIA Jetson Xavier is connected to a monitor for human detection through the Real-Time Messaging Protocol (RTMP) (**b**)

All the algorithms were implemented and executed on Tensorflow 2. TensorFlow [53], which is a state-of-the-art machine learning platform, is fully compatible with NVIDIA CUDA; therefore, it is the selected platform for implementation and execution of the algorithm.

OpenCV [54] is also deployed for the purpose of image processing due to being mature in the field of computer vision.

5 Results and discussion

A set of videos were collected from the crane site to validate and further evaluate the effectiveness of the proposed model against different state-of-the-art models.

5.1 Quantitative results

Table 4 shows the comparison results of various state-of-the-art models including YOLOv4 the improved version of YOLOv3 [44] against the Smart Crane approach (our approach).

5.2 Accuracy and speed comparison

According to the results in Table 4 and Fig. 8, the standard YOLOv4 achieved a high accuracy of 93.62% but with 6.1 FPS on the Jetson Xavier small-form PC with input size of 608 which is not suitable for our use case. Hence, the standard Tiny-YOLOv4 [44], which is a simplified, light version of YOLOv4, was also compared with the CraneNet approach. It achieved an accuracy of 90.0% with 31 FPS on the Jetson Xavier. Standard Tiny-YOLOv3 was also trained and compared. The results show that the accuracy is less (86.0%) with 29 FPS and input size of 608 on the Jetson Xavier. However, they only use two output layers compared with three output layers in standard YOLOv4, which decreases the accuracy of human detection at higher altitudes (> 20 m). Hence, an extra output layer has been added to the standard Tiny-YOLOv4 [44]. Tiny-YOLOv4 with 3 layers (3l) achieved an accuracy of 91.5% with FPS of 28. Our approach achieves an accuracy of 92.59% and 19 FPS on the Jetson Xavier. Considering both speed and accuracy, the Smart Crane approach has high accuracy on top-view small human detection, which is an imperative factor when it comes to workforce safety and still fulfil the speed requirement for our use case.

5.3 Model size

Table 4 and Fig. 9 show the average model size versus the average Billion Floating-Point Operations (BFLOPS) of the concerned models. According to the results, the

Table 4 Accuracy, speed, model size and BPLOPS of different models

Index	Model	Accuracy (mAP)	Speed (FPS)	Model size (MB)	BFLOPS
1	Standard Tiny-YOLOv4	90	31	23.5	14.5
2	Standard Tiny-YOLOv3	86.0	29	34.7	11.6
3	Standard Tiny-YOLOv4(3l)	91.5	28	24.5	17.1
4	Standard YOLOv4	93.62	6.1	256	127.2
5	Our approach	92.59	19	42.9	24.5

Table 5 Accuracy and speed of different models with SDD dataset

Model	Input size	Pedestrian (mAP)	Biker (mAP)	Car (mAP)	Accuracy (mAP)	Speed (FPS)	GPU
Standard YOLOv3	416 × 416	73.52	51.33	61.8	62.23	46.94	TITAN X
PeleeNet	304 × 304	68.10	59.02	74.26	67.13	30.21	TITAN X
PeleeNet +DMS	304 × 304	69.27	60.14	75.44	68.28	23.60	TITAN X
Our approach	304 × 304	60.28	58.66	77.13	65.35	93.2	TITAN X
Our approach	416 × 416	61.25	60.04	80.60	67.29	75.8	TITAN X
Our approach	416 × 416	61.25	60.04	80.60	67.29	26.5	Jetson
Our approach	608 × 608	67.50	63.34	79.61	70.15	19	Jetson

standard Tiny-YOLOv4 had the smallest model size and the BFLOPS of 23.5 and 14.5, respectively.The standard Tiny-YOLOv3 has the model size and the BFLOPS of 34.7 and 11.6. The standard Tiny-YOLOv4 (3l) had the model size and the BFLOPS of 24.5 and 17.1, respectively. The model size of CraneNet is (42.9 MB) with 24.5 BFLOPS. The standard YOLOv4 had the model size of 256 and 127.2 BFLOPS which is not light enough for our use case on the Jetson Xavier. Our approach, on the other hand, is highly accurate whilst still maintaining the trade-off between accuracy and speed.

5.4 Results at different altitude

The accuracy of the proposed model was also evaluated at different altitudes, and the accuracy was 93.2, 94.3, 93.4, 92.0 and 90.1 at the altitudes of 10, 20, 30, 40 and 50 meters, respectively.

5.5 Results with public dataset

Since our collected dataset is private, we have also compared the performance of our model with two public datasets suitable for our use case. For this purpose, the Standford Drone Dataset (SDD) [55] and Visdrone 2019 dataset [56] were chosen for this purpose. The SDD dataset was chosen due to images being taken from top-view perspective. The SDD Dataset is a very large-scale dataset taken in a real-world university campus with six classes of objects. Since the orthoimagery contains limited information for object detection, we used a subset of it with four

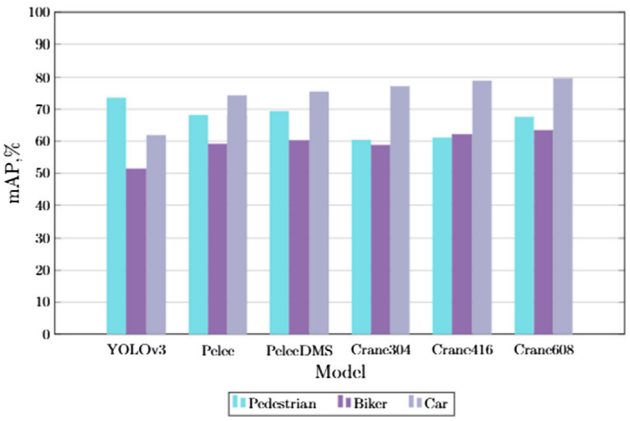


Fig. 6 mAP for pedestrian, biker and car objects using SDD dataset. The turquoise bar represents the pedestrians. The purple bar shows the bikers, and the gray indicates the car. The accuracy of pedestrians, bikers and cars was 73.52%, 51.33% and 61.8% with an input size of 416 for YOLOv3. The accuracy of pedestrians, bikers and cars was 68.10%, 59.02% and 74.26% for PeleeNet. The accuracy of pedestrians, bikers and cars was 69.27%, 60.14% and 75.44% for PeleeNet+DMS. The accuracy of pedestrians, bikers and cars was 60.28, 58.66, 77.13 for CraneNet with the input size of 304, 61.25%, 60.04% and 80.60% with input size of 416 and 67.50%, 63.34% and 79.61% with the input size of 608

scenes, named bookstore, hyang, death circle and little, respectively [57]. The videos were shortened to videos of 1-min length by removing the repetitive, unuseful frames. Totally 42,462 frames were extracted from the aforementioned scenes (bookstore (12,304), death circle (5,513), hyang (18,404) and little (6,247)). Three classes of objects (pedestrian, Biker and car) were selected for comparison. The Visdrone 2019 dataset as a second dataset was selected

Fig. 7 Example of human detection at different altitudes in an industrial operational site. The images on the first row show the human detection up to 10 m; images on the second row show the human detection up to 20 m. The third row displays the human detection up to 30 m, and the last row illustrates the human detection up to 50m

due to images were captured from a drone and at distance and thereby being small in size and suitable to be used to further prove the robustness of our proposed model. The Visdrone 2019 dataset was collected and manually annotated by the AISKYEYE with 288 videos (261,908 frames and 10,209 static images). The images covers various scenarios in 14 various cities containing different objects such as pedestrian, vehicles, bicycles under different weather and lighting conditions with more than 2.6 million bounding boxes.

The CraneNet was compared with Standard YOLOv3 [20], PeleeNet [58], and PeleeNet+DMS [57] with Deep Motion Saliency (DMS) [58] using SDD. Apart from

Table 6 Comparison of results of our approach and SlimYOLOv3 on Visdrone 2019

Models	Input	mAP (%)	FPS	Model size (MB)
SlimYOLOv3	416	15.70	67.0	79.6
CraneNet	416	24.44	75.8	47.8

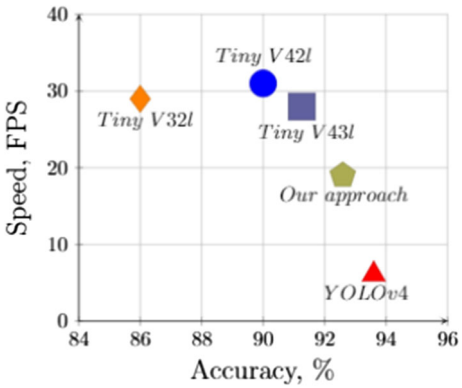


Fig. 8 Accuracy versus Speed. The standard YOLOv4 achieved a high accuracy of 93.62% but with 6.1 FPS. Tiny-YOLOv4 achieved 90.0% with 31 FPS on the Jetson Xavier. Standard Tiny-YOLOv3 has the accuracy of (86.0%) with 29 FPS. Tiny-YOLOv4 with 3 layers (3l) achieved an accuracy of 91.5% with FPS of 28. Our approach obtained an accuracy of 92.59% and 19 FPS on the Jetson Xavier

Jetson Xavier and for the sake of comparison, we also evaluated the result of our approach with the input size of 304 and 416 on TITAN X Platform (a computer with an NVIDIA TITAN X GPU with 12 GB RAM), since the results of compared models have been evaluated on TITAN X GPU in literature [43] [20].

Regarding the standard YOLOv3 being trained with SDD, based on Table 5 and Fig. 6, the average accuracy of 62.23% was obtained with a speed of 46.94 FPS on an NVIDIA TITAN X (Pascal) GPU device [58]. The accuracy of pedestrians, bikers and cars was 73.52%, 51.33% and 61.8% with an input size of 416 for YOLOv3. As apparent from the results, the overall accuracy is lower with an input size of 416 (62.23% against 67.29%). In addition, YOLOv3 introduces high power consumption and computational overhead to embedded devices such as a Jetson Xavier [59] with FPS of around 10 with the input size of 416 which is slow and unsuitable for our use case. In terms of PeleeNet, the accuracy of pedestrians, bikers and cars was 68.10%, 59.02% and 74.26%. Although the overall accuracy of 67.13% was achieved, which is slightly higher than that of CraneNet (65.35%) with the input size of 304, the FPS of 30.21 was obtained on an NVIDIA TITAN X (Pascal) GPU device [58] that is three times less than our approach on TITAN X and thereby slow and

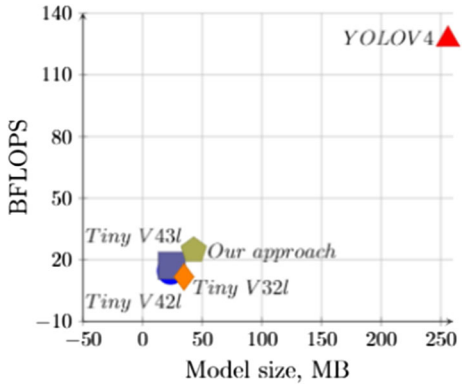


Fig. 9 Model size versus BFLOPS. The standard Tiny-YOLOv4 had the model size and the BFLOPS of 23.5 MB and 14.5. The standard Tiny-YOLOv3 has the model size and the BFLOPS of 34.7 MB and 11.6. The standard Tiny-YOLOv4 (3l) had the model size and the BFLOPS of 24.5 MB and 17.1. The model size of CraneNet is 42.9 MB with 24.5 BFLOPS. The standard YOLOv4 had the model size of 256 MB and 127.2 BFLOPS

computationally expensive for our use-case on Jetson. Similarly, PeleeNet+DMS model achieved 23.60 FPS on a TITAN X GPU device with an input size of 304, which is more than 4 times lower than our approach with 93.2 FPS on TITAN X that makes it slow and unsuitable for our use-case on Jetson. Although the accuracy is higher (68.28% against 65.35%). The accuracy of pedestrians, bikers and cars was 69.27%, 60.14% and 75.44% for PeleeNet+DMS. The accuracy of pedestrians, bikers and cars was 61.25 %, 62.04% and 80.60% for CraneNet with input size of 416, and 67.50%, 63.34% and 79.61% with the input size of 608. In terms of CraneNet approach, the results demonstrate that the proposed model can achieve the best accuracy and speed trade-off which was imperative for the success of the project.

To further prove the robustness of the proposed algorithm apart from Standford dataset, it was also trained with Visdrone 2019 and compared to SlimYOLOv3 [60] as an accurate real-time model suitable for UAV applications and small objects. According to Table 6, our approach has more accuracy than SlimYOLOv3 (24.44% versus 15.70%) and faster (75.8 FPS versus 67 FPS) on TITAN X with the input of 416.

5.6 Qualitative results

The detection results of our approach have been shown at different altitudes up to 50 meters at different industrial sites. To achieve these results, separate unseen testing videos were taken at different construction sites in real scenarios, in a complex, cluttered environment. Figure 7 illustrates some results obtained in situ. As apparent from the results, our approach had a successful detection of workforce on the crane operational site at different

altitudes up to 50 meters with no false negative or positive detection in all the scenarios tested.

6 Concluding remarks and future work

The industrial operations of cranes can be hazardous to workforce due to the operator's lack of visibility and can cause accidents and even fatalities on industrial sites. The proposed Smart Crane solution reduces these hazards by monitoring and detecting the workforce with high accuracy from top-view and enables the operator to take concrete and swift actions on workforce safety. The solution was implemented on an embedded system (Jetson Xavier) and thus is highly space friendly and portable to be deployed in operational environments. With TensorFlow 2 as the machine-learning platform for the proposed Smart Crane model (CraneNet) to be executed on, the system was optimised for detecting humans up to 50 meters of altitude, and it has achieved 92.59% of accuracy at 19 FPS on the power-constrained device (Jetson Xavier). Taking into account a trade-off between accuracy and speed, the proposed approach shows the best performance in small object detection compared with state-of-the-art approaches. Moreover, the CraneNet has been trained with the SDD dataset and Visdrone 2019 due to containing top-view and small-sized human objects in the images and compared it with other architectures trained with this model. The results have proved the efficacy of the proposed model on resource-constrained platforms. The proposed system has improved the safety of workforce by minimising the risks of crane operations on industrial sites and can be employed for other objects detection from top-view on industrial sites. As future work, an all-day (day and night) accurate human detection system will be designed, implemented and expanded at more complex and cluttered industrial operational sites. For this purpose, the crane will also be equipped with a thermal camera to capture images at twilight, dusk, and night. In addition, the future work will include the detection of humans in adversarial weather conditions, such as foggy and rainy, in which visibility is low and makes the small object detection even harder. In addition, the proposed novel model will also be trained for other objects including animals as intruders to industrial sites.

Acknowledgements This work was in part funded by the Innovation Centre for Sensor and Imaging Systems (CENSIS) and in collaboration with Intebloc, under the grant number CAF-655 ("SmartCrane: Real-time Human Detection System for Safe Crane Operations"). The authors would like to thank all the partners in this project for their support.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Roberts D, Bretl T, Golparvar-Fard M (2017) Detecting and classifying cranes using camera-equipped uavs for monitoring crane-related safety hazards
2. Yang Z et al (2019) Safety distance identification for crane drivers based on mask R-CNN. Sensors 19(12):2789
3. Sutjaritvorakul T, Vierling A, Berns K (2020) Data-Driven Worker Detection from Load-View Crane Camera. In: ISARC proceedings of the international symposium on automation and robotics in construction, vol. 37. IAARC Publications, pp. 864–871
4. Neuhausen M, Teizer J, König M (2018) Construction worker detection and tracking in bird's-eye viewcamera images. In: ISARC proceedings of the international symposium on automation and robotics in construction, vol. 35. IAARC Publications, pp.1–8
5. Martinez-Alpiste I, Golcarenarenji G, Wang Q, Alcaraz Calero JM (2020) Altitude-adaptive and cost-effective object recognition in an integrated smartphone and uav system. In: 2020 European Conference on Networks and Communications (EuCNC), pp 316–320. https://doi.org/10.1109/EuCNC48522.2020.9200951
6. Martinez-Alpiste I, Golcarenarenji G, Wang Q, Calero JA (2020) Real-time low-pixel infrared human detection from unmanned aerial vehicles. In: Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications
7. Martinez-Alpiste I, Golcarenarenji G, Wang Q, Alcaraz-Calero JM (2021) Search and rescue operation using UAVs: a case study. Expert Syst Appl 178:114937
8. Qi W, Aliverti A (2020) A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity. IEEE J Biomed Health Inform 24:2199–2207
9. Qi W, Su H, Aliverti A (2020) A smartphone-based adaptive recognition and real-time monitoring system for human activities. IEEE Trans Hum Mach Syst 50:414–423
10. Ahmad M, Ahmed I, Ullah K, Khan I, Khattak A, Adnan A (2019) Person detection from overhead view: a survey. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2019.0100470
11. Rubaiyat AHM, Toma TT, Khandani MK, Rahman SA, Chen L, Ye Y, Pan C (2016) Automatic detection of helmet uses for construction safety. In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW) pp 135–142
12. Seong H, Choi H, Cho H, Lee S, Son H, Kim C (2017) Vision-based safety vest detection in a construction scene. In ISARC

proceedings of the international symposium on automation and robotics in construction, vol. 34. IAARC Publications
13. Hu J, Gao X, Wu H, Gao S (2019) Detection of workers without the helments in videos based on yolo v3. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp 1–4. https://doi.org/10.1109/CISP-BMEI48845.2019.8966045
14. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision 2015 Inter. pp 1440–1448. https://doi.org/10.1109/ICCV.2015.169
15. Ren S, He K, Girshick R, Sun J (2017) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
16. Zeng L, Sun B, Zhu D (2021) Underwater target detection based on faster r-cnn and adversarial occlusion network. Eng Appl Artif Intell 100:104190
17. He K, Gkioxari G, Dollár P, Girshick R (2020) Mask r-cnn. IEEE Trans Pattern Anal Mach Intell 42(2):386–397. https://doi.org/10.1109/TPAMI.2018.2844175
18. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem. pp 779–788 https://doi.org/10.1109/CVPR.2016.91
19. Redmon J, Farhadi A (2017). YOLO9000: better, faster, stronger. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-Janua. pp 6517–6525. https://doi.org/10.1109/CVPR.2017.690
20. Redmon J, Farhadi A (2018) YOLOv3: An incremental improvement http://arxiv.org/abs/1804.02767
21. Van Etten A (2018). You only look twice: rapid multi-scale object detection in satellite imagery. http://arxiv.org/abs/1805.09512
22. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
23. Krishna H, Jawahar CV (2017) Improving small object detection. In: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR) pp 340–345
24. Lin TY, Dollár P, Girshick RB, He K, Hariharan B, Belongie SJ (2017) Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp 936–944
25. Golcarenarenji G, Martinez-Alpiste I, Wang Q, Alcaraz-Calero JM (2020) Efficient real-time human detection using unmanned aerial vehicles optical imagery. Int J Remote Sens. https://doi.org/10.1080/01431161.2020.1862435
26. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
27. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824
28. Maheshwari P, Alex D, Banerjee S, Behera S, Panda S (2018) Top view person detection and counting for low compute embedded platforms. In: Proceedings of the 2018 the 2nd International Conference on Video and Image Processing. pp 35–43
29. Ahmed I, Ahmad M, Khan FA, Asif M (2020) Comparison of deep-learning-based segmentation models: using top view person images. IEEE Access 8:136361–136373. https://doi.org/10.1109/ACCESS.2020.3011406
30. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965
31. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. http://arxiv.org/abs/1505.04597
32. Chen L.C., Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. http://arxiv.org/abs/1706.05587
33. Ahmad M, Ahmed I, Khan FA, Qayum F, Aljuaid H (2020) Convolutional neural network-based person tracking using overhead views. Int J Distrib Sens Netw 16(6):1550147720934738
34. Sutjaritvorakul T, Vierling A, Pawlak J, Berns K (2020) Simulation platform for crane visibility safety assistance. In: Zeghloul S, Laribi MA, Sandoval Arevalo JS (eds) Advances in Service and Industrial Robotics. Springer International Publishing, Cham, pp 22–29
35. Ahmad M, Ahmed I, Adnan A (2019) Overhead view person detection using yolo. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON). pp 0627–0633. https://doi.org/10.1109/UEMCON47517.2019.8992980
36. Ullah K, Ahmed I, Ahmad M, Rahman AU, Nawaz M, Adnan A (2019) Rotation invariant person tracker using top view. J Ambient Intell Humaniz Comput 1–17
37. Ahmed I, Ahmad M, Adnan A, Ahmad A, Khan M (2019) Person detector for different overhead views using machine learning. Int J Mach Learn Cybern 10(10):2657–2668
38. Yamamoto J, Inoue K, Yoshioka M (2017) Investigation of customer behavior analysis based on top-view depth camera. In: 2017 IEEE Winter Applications of Computer Vision Workshops (WACVW) pp 67–74
39. Ahmad M, Ahmed I, Ullah K, Khan I, Adnan A (2018). Robust background subtraction based person's counting from overhead view. In: 2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON). pp 746–752. https://doi.org/10.1109/UEMCON.2018.8796595
40. Ahmed I, Adnan A (2017) A robust algorithm for detecting people in overhead views. Clust Comput 21:633–654
41. Martinez-Alpiste I, Casaseca-de-la-Higuera P, Alcaraz-Calero J, Grecos C, Wang Q (2019). Benchmarking machine-learning-based object detection on a uav and mobile platform. In: 2019 IEEE Wireless Communications and Networking Conference (WCNC). pp 1–6. https://doi.org/10.1109/WCNC.2019.8885504
42. Martinez-Alpiste I, de-la Higuera PC, Alcaraz-Calero JM, Grecos C, Wang Q (2020) Smartphone-based object recognition with embedded machine learning intelligence for unmanned aerial vehicles. J Field Robot 37:404–420
43. Jiang Z, Zhao L, Li S, Jia Y (2020) Real-time object detection method based on improved yolov4-tiny. http://arxiv.org/abs/2011.04244
44. Bochkovskiy A, Wang C.Y., Liao H (2020) Yolov4: Optimal speed and accuracy of object detection. http://arxiv.org/abs/2004.10934
45. AlexeyAB: Darknet (2020b) https://github.com/AlexeyAB/Yolo_mark
46. Dwibedi D, Misra I, Hebert M (2017) Cut, paste and learn: surprisingly easy synthesis for instance detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp 1301–1310
47. Jung AB, Wada K, Crall J, Tanaka S, Graving J, Reinders C, Yadav S, Banerjee J, Vecsei G, Kraft A, Rui Z, Borovec J, Vallentin C, Zhydenko S, Pfeiffer K, Cook B, Fernández I, De Rainville FM, Weng CH, Ayala-Acevedo A, Meudec R, Laporte M, et al. (2020) Data augmentation https://github.com/aleju/imgaug

48. Perez L, Wang J (2017) The effectiveness of data augmentation in image classification using deep learning http://arxiv.org/abs/1712.04621
49. AlexeyAB: Darknet (2020a) https://github.com/AlexeyAB
50. Loshchilov I, Hutter F (2019) SGDR: stochastic gradient descent with warm restarts. In: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings pp 1–16
51. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 5:7
52. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision - ECCV 2014. Springer International Publishing, Cham, pp 740–755
53. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado G.S., Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems https://www.tensorflow.org/. Software available from tensorflow.org
54. Bradski G (2000) The OpenCV Library. Dr. Dobb's Journal of Software Tools
55. Robicquet A, Sadeghian A, Alahi A, Savarese S (2016) Learning social etiquette: human trajectory understanding in crowded scenes. In: ECCV
56. Du D, Zhu P, Wen L, Bian X, Lin H, Hu Q, Peng T, Zheng J, Wang X, Zhang Y, et al. (2019) VisDrone-DET2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp 0–0
57. Zhang J, Liang X, Wang M, Yang L, Zhuo L (2020) Coarse-to-fine object detection in unmanned aerial vehicle imagery using lightweight convolutional neural network and deep motion saliency. Neurocomputing 398:555–565
58. Wang R, Li X, Ao S, Ling C (2018) Pelee: a real-time object detection system on mobile devices. In: NeurIPS
59. Mao Q, Sun H, Liu Y, Jia R (2019) Mini-yolov3: real-time object detector for embedded applications. IEEE Access 7:133529–133538. https://doi.org/10.1109/ACCESS.2019.2941547
60. SlimYOLOv3 (2019) Narrower, faster and better for real-time uav applications. In: Zhang P, Zhong Y, Li X 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW) pp 37–45