Periodica Polytechnica Electrical Engineering and Computer Science, 67(1), pp. 102–111, 2023

Object Classification and Tracking Using Scaled P8 YOLOv4 Lite Model

Shakil Shaikh[1]*, Jayant Chopade[1], Gajanan Kharate[1]

[1] Department of Electronics and Telecommunication, Matoshri College of Engineering & Research Centre, Eklahare, Nashik, Savitribai Phule Pune University, 422105 Maharashtra, P.O.B. 411007, India

* Corresponding author, e-mail: shakils68@rediffmail.com

Received: 20 June 2022, Accepted: 02 November 2022, Published online: 07 December 2022

Abstract

One of the most difficult tasks in the area of computer vision is object detection, which combines object categorization and object location within a scene. In terms of object detection, Deep Neural Networks have been recently demonstrated to outperform alternative approaches. The issues related deep learning neural network is its complexity and huge computation, so it is not possible to detect and track the objects in image of high resolution in real time. We proposed scaled YOLOv4 lite model as Single Stage Detector Neural Network for object detection, tracking and it is trained using COCO 2017 dataset. To create the YOLOv4-CSP- P5- P6- P7- P8 networks, the Scaled YOLOv4 applied efficient network scaling strategies. The additional layer in YOLOv4 lite model is added as P8 layer which improves accuracy. Cross-stage-partial (CSP) connections and Mish activation are used in improved network design, such as backbone optimization and Neck (PAN). In the case of YOLOv4, however, it can only be trained once for all resolutions. Width and Height activations have been changed, allowing for faster network training. With YOLOv4 lite model, we used CSPDarkNet-53 model as a backbone. The experimental result show our YOLOv4 lite model can detect and track object up to 28 fps when model run with the video input and has an accuracy of 86.09% when tested on real-time video with resolutions 1920 × 1080 (full HD). AP = 50.81%, AP @50 = 63.6%, and AP @75 = 52.5% for CSPDarkNet-53 model backbone.

Keywords

cross stage partial, object detection, computer vision, Deep Neural Network, backbone

1 Introduction

Object recognition and classification is critical at the country's border, where security is paramount. Object detection, tracking, and identification can be accomplished using a variety of approaches. However, utilizing image processing to reliably identify things in real time is a difficult task. The method of recognizing main things in an image is known as salient object detection. Humans can quickly recognize which objects are important in a scenario. The goal of automating this process is to produce machines that can accurately simulate human. To recognize prominent objects, background subtraction and the Gabor filter are used [1]. The most basic implementation of background subtraction is to pixel-wise evaluate the difference between a previously taken or estimated background image and the current image, and then threshold the difference value to find the pixels that belong to moving objects. Due to its ease of development, low processing costs, and lack of prior knowledge of the target objects, this simple background removal is used by many applications. This technique, however, has a number of significant drawbacks. It can't adjust to dynamic background changes like fluttering leaves, changing lighting, or camera movement, for example. Second, when moving objects and the background have color combinations, the detection performance degrades. These shortcomings frequently result in false positives and negatives, respectively [2].

In harvesting robot platforms, fruit detection is critical. Fruit recognition has proven difficult due to complex environmental factors such as lighting variation and occlusion. To overcome detection challenges, a strong YOLO-Muskmelon scheme that is both accurate and quick was presented [3]. The task of object recognition and classification is crucial in computer vision. YOLO is an extremely effective method for recognition and classification. Hardware architectures for running YOLO model in real-time on embedded hardware has been considered [4]. Deep learning-based object detection methods are classified

Cite this article as: Shaikh, S., Chopade, J., Kharate, G. "Object Classification and Tracking Using Scaled P8 YOLOv4 Lite Model", Periodica Polytechnica Electrical Engineering and Computer Science, 67(1), pp. 102–111, 2023. https://doi.org/10.3311/PPee.20685

into two types: region proposal-based two-stage methods and regression-based one-stage methods. The region-based convolution neural network (R-CNN) approach is a common two-stage method. Fast R-CNN, Faster RCNN, fully convolutional networks with a focus on a region of interest (R-FCN), light head R-CNN, and various convolution neural network enhance methods are all available. Although, the two-stage method is more accurate than the one-stage method, the one-stage method has a faster detection speed. In some situations where real-time processing is required, the one-stage technique is preferable. Redmon et al. [5] proposed the YOLO approach, which is based regression-based one-stage method. He also presented the YOLOv2, which is built on YOLO and involves eliminating the completely connected layer and the last pooling layer, using anchor boxes to forecast bounding boxes, and creating a new basic network called DarkNet-19. The YOLOv3 is the most recent development in YOLO approach. To increase detection accuracy and the capacity to recognize tiny objects, it incorporates a feature pyramid network, a better basic network termed DarkNet-53, and binary cross-entropy loss.

YOLOv4 model recognizes object in real-time, which was launched in April 2020 and reached state-of-the-art performances on the Data set. It works by dividing the object identification process into two parts: regression for determining object location via bounding boxes, and classification for determining the item's class. The DarkNet-53 framework is used in this YOLOv4 implementation. In single-stage detector models, the classes and bounding boxes for the full image are predicted rather than choosing the region of interest. They are therefore faster than two-stage detectors as a result. On the other hand the You Only Look Once (YOLO) framework approaches object identification in a different way. The bounding box coordinates and class probabilities for these boxes are predicted by taking the entire image into account in a single instance. Therefore, YOLO framework speed is enhanced while maintaining accuracy that is nearly identical to two stage technique. In contrast, two-stage object detection splits the object identification problem into two parts: identifying potential object regions of interest and classifying the image. Compared to YOLO, it takes longer to detect objects.

The real-time pattern recognition could distinguish many objects from a single image, frame a confined-edge box around nearby objects, and train and deploy in a production system in a short amount of time. A new approach for detecting small objects in high-resolution photos has been proposed. The size of the convolutional layer and the resolution of the input image determine the number of pyramid layers. To enhance the accuracy of detecting small objects, breaking the overlapping blocks on each layer of the pyramid except the top one. If two detected regions belong to the same class and have a high overlapping value, they are merged into one. In high-definition video, the method outperforms YOLOv4 in recognizing small objects [6]. Object detection applications require the ability to interpret data in real time in high-resolution monitoring systems. Real-time moving object recognition is difficult to do due to the vast the volume of data needed for high-resolution images. There are numerous hurdles, including complicated backgrounds, varying lighting, local motion from moving trees or items hidden by dust. The resolution of recording devices is steadily improving, necessitating the development of new ways for processing high-resolution data for Object detection and tracking from static image or video.

1.1 Research contributions

We presented an innovative technique for processing high-resolution video data that keeps a balance between accuracy and speed performance. We implemented a scaled YOLOv4 lite model for Object detection and tracking from static image or video. The model was trained on the COCO 2017 dataset. To create the YOLOv4-CSP-P5-P6-P7-P8 networks, the Scaled YOLOv4 applied efficient network scaling strategies. The P8 layer is incorporated as an extra layer in the YOLOv4 light model, which enhances accuracy especially for small object detection. During training, the Exponential Moving Average (EMA) is employed for weight-averaging. The neural network must be trained independently for each resolution, whereas YOLOv4 simply needs to be trained once for all resolutions. Enhanced normalizer is employed in YOLO layers. Width and Height activations have been changed, allowing for faster network training. The CSP DarkNet-53 model was combined with the YOLOv4 light model.

When compared to ResNet-based architectures, the CSPDarkNet-53 backbone has a greater accuracy in object recognition while also having a superior categorization performance. It is more suitable to real-time object identification, especially for embedded device development. After scaling the proposed target detection method, the subsequent stage is to address the quantitative features that will vary, such as the number of parameters with qualitative factors. Model inference time, average accuracy, and other characteristics are among them. Qualitative elements will have varying gain effects depending on the equipment or database used. The model was trained on a desktop PC with a processor i7 in Google Colab and evaluated on a free GPU given by Colab.

2 Related work

Srivastava and Srivastava [1] employed background removal, Gabor filters, objectness, and minimum directional backgroundness to tackle the problem of significant object recognition. Researchers haven't looked at deep learning techniques. Hosaka et al. [2] offer a method for identifying moving objects using a Markov random field (MRF) model. The goal is to resolve two major problems in previous methods: false positives from dynamic background changes like fluctuation trees and camera motion, and false negatives from the actuality of similar colors in objects and their backgrounds. Lawal [3] demonstrated the YOLOMuskmelon model, which included a ReLU activated ResNet43 backbone, a new 2,3,4,3,2 residual block arrangement, spatial pyramid pooling (SPP), Complete Intersection over Union (CIoU) loss, feature pyramid network (FPN), and Distance Intersection over Union-based distance. Pestana et al. [4] proposed a core that is configured for real-time YOLOv3 and YOLOv4-Tiny execution, is implemented in a RISC-V-based system-on-chip architecture, and prototyped in FPGA (Field Programmable Gate Array). By merging background subtraction with Convolutional Neural Networks, Redmon et al. [5] introduce YOLO, a unified model for object detection. The provided model is simple to build and can be immediately trained on the full frame. Zhu et al. [6] explored real-time object detection in high-resolution video frames with remarkable accuracy. Moving object detection employing Background subtraction and a blend of Gaussian algorithms was proposed by HariPriya and Aman [7]. Moving object detection is done with a background subtraction approach, while moving object classification is done with a blend of Gaussian algorithms. In order to address the issues with floating-point-based quantization methods for YOLOv3 and YOLOv4, Kim and Kim [8] suggested a fixed-point-based quantization technique tailored for embedded platforms. An enhanced penalty function based on the Complete Intersection over Union loss function is suggested by Wang et al. [9] to boost the positioning accuracy.

Sambolek and Ivasic-Kos [10] investigated state-of-the-art person detectors in drone photos and presented a model for detecting people during SAR operations. CNN-based object detectors, including the Cascade R-CNN, Faster R-CNN, RetinaNet, and YOLOv4, were trained and evaluated on selected drone images. Bhatti et al. [11] demonstrated a unique real-time automatic weapon detection system. The work does undoubtedly aid in enhancing security, law and order for the betterment. Maddalena and Petrosino [12] described a detailed assessment of algorithms that use RGBD data for object recognition based on background removal, which is a basic block for many computer vision applications. Shaikh et al. [13] presented a common technique background subtraction for accurately detecting moving objects in videos taken by stationary cameras. Zheng et al. [14] offer an enhanced YOLOv3 network model and build a large-scale bearing-cover defect dataset. Wang et al. [15] addressed the issue of image background and foreground imbalance by incorporating the foreground and background balance loss function into the YOLOv4 loss function component. Srivastava et al. [16] compared the most recent and sophisticated CNN-based object detection techniques and concluded thatYOLOv3 shows the best overall performance. In order to categorize different sorts of crises and deduce the necessary emergency measures, Asif et al. [17] offered an automatic analysis of social media images. The YOLOv2 Convolutional Neural Network was used by Saponara et al. [19] to demonstrate real-time video-based fire and smoke detection in antifire surveillance systems. To account for the needs of embedded devices, YOLOv2 is built with a light-weight neural network architecture. In order to enhance the framework for managing garbage in smart cities, Kumar et al. [20] developed a novel YOLOv3 algorithm application for garbage segregation and YOLOv4 for detection of tomato [21]. To address the issues of low accuracy, low real-time performance and other issues, the modified YOLO-v4-based face mask identification [22] method were utilized. New architectures of YOLO for Vehicle License Plate Detection is proposed [23]. Wu et al. [24] developed an enhanced YOLO v4 and data augmentation techniques to help the apple picking robot. Schütz et al. [25] proposed using YOLOv4 to detect and monitor Red Foxes' movements.

Based on the YOLOv4 algorithm, Lee and Lin [26] and Janardhan et al. [27] proposed using the sense of hearing to view an object held in front of a person and a camera. Růžička and Franchetti [28] suggested an attention pipeline approach that limits the total number of necessary assessments by using two staged evaluations of each image or video frame under rough and refined resolution. Convolutional Neural Networks are used by Ammar et al. [29] to solve the problem of car recognition from aerial images. Cao et al. [30] presented face mask object detection in real time. Liu et al. [31], proposed a YOLOv4-based model for sea surface object detection scheme. Ma et al. [32] reported an improved YOLOv4 small algorithm. The simulation results reveal that, when compared to YOLOv4-tiny, the upgraded network structure has a 3.3% higher accuracy and a detection speed of 251 frames per second, which meets the real-time detection criteria. Liu et al. [33] suggested a CNN-based technique for detecting tiny drones. High Resolution Low-latency Block-wise Object Recognition Method Using SSD was introduced by Magalhães et al. [34]. Using certain pre-existing adjustments and methodologies, Mahto et al. [35] fine-tuned the new state-of-the-art object recognition system YOLOv4 to especially fit the needs of vehicle detection. Bochkovskiy et al. [36] presented YOLOv4 for Optimal Speed and Accuracy in Object Detection.

Kim et al. [37] established a hybrid framework to detect and recognize moving objects. A new object detection model, YOLOv4-FPM, is proposed by Yu et al. [38] for real-time detection. Zhang et al. [39] offer Smart-YOLO, a new real-time object detection system. It has a small model magnitude, a rapid detection speed, and is better suited to being promoted to some edge or mobile end devices. The technique proposed by Bohusha et al. [40] for recognizing objects in 4K and 8K images and it has a great efficiency in recognizing small objects in 4K and 8K quality images. Kadadi et al. [41] shows how to use the background subtraction (BGS) method to find and follow the intended moving objects (MOs). The BGS method offers the possibility of cost savings because data storage begins as soon as motion is detected. Effective MO detection was the goal of the BGS method.

Summary of Literature Review: Even though researchers developed a variety of background subtraction techniques, there are still certain problems, such as accuracy and speed in high resolution images. The researchers suggested a few methods for object detection, such as YOLOv3 and YOLOv4, which can increase speed but have relatively low recall and accuracy. They also struggle to recognise small objects and close objects because each grid can only suggest two bounding boxes. The author suggests the SSD or Fusion technique for object detection in high resolution, but the model will still not run in real time to detect the objects. So we proposed object classification and tracking using single stage detector, improved YOLOv4 in Full HD Video.

3 Materials and methods

Fig. 1 shows proposed architecture for Scaled YOLOv4 with P8 for object detection from static image and video.

In the proposed architecture the following are the abbreviation used:

1. A = conv $k = 3$, $s = 1$;
2. B = (conv $k = 3$, $s = 2$) → (m × CSP);
3. C = (rCSP) → (SPP) → (rCSP) → (rCSP);
4. D = (conv $k = 1$, $s = 1$) → (up s = 2) }
   (conv $k = 1$, $s = 1$) } (Concat) → (m × rCSP);
5. E = (identity)
   (Conv $k = 3$, $s = 2$) } (Concat) → (m × rCSp);
6. F = (conv $k = 3$, $s = 1$) → (YOLOv4).

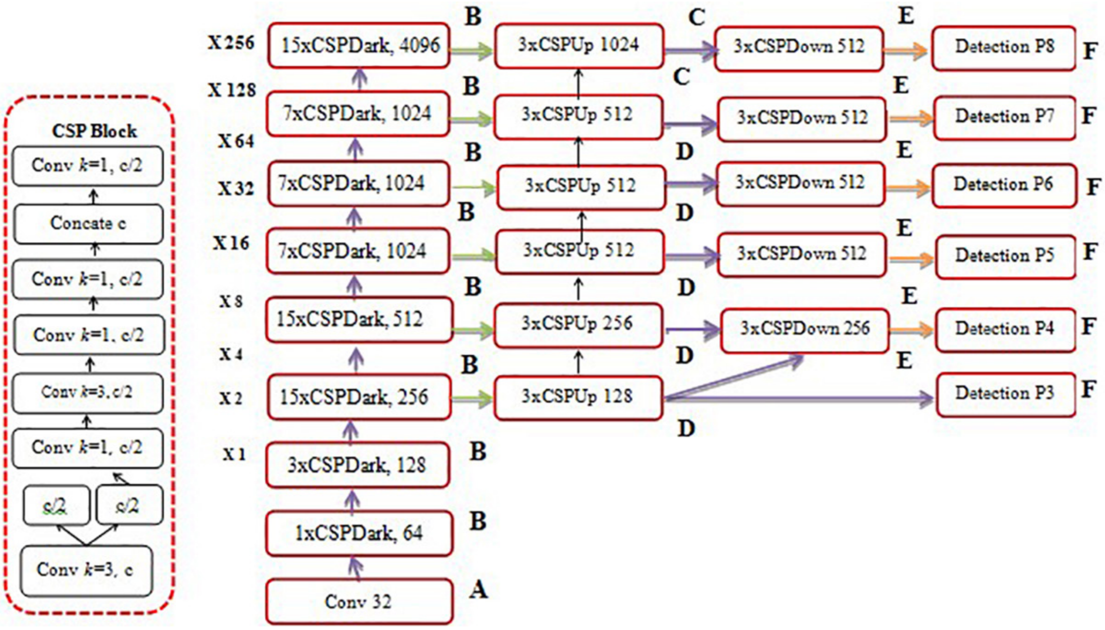

Fig. 1 Scaled YOLOv4 with P8 architecture for object detection

3.1 Improved scaled YOLOv4 architecture

YOLOv4 scaled to create the YOLOv4-CSP-P5-P6-P7-P8 networks, YOLOv4 applied with efficient network scaling approaches. The Neck is employed Cross-stage-partial (CSP) connections and Mish activation, and the backbone has been optimized. During training, the Exponential Moving Average (EMA) is employed for weight-averaging. The neural network must be trained independently for each resolution, whereas YOLOv4 simply needs to be trained once for all resolutions. Enhanced normalizer is employed in YOLO layers. Width and Height activations have been changed, allowing for faster network training.

Changing the depth of the model, that is, adding more convolutional layers is the standard model scaling strategy. Simonyan et al. built VGGNet, which superimposed additional convolutional layers at various stages and used this notion to design the VGG11, VGG13, VGG16, and VGG19 designs. The model scaling method is typically used in subsequent methods. In the ResNet extended depth can be employed to build very deep networks, such as ResNet-50, ResNet-101, and ResNet-152. Later, altered the amount of convolutional layer cores to accommodate for the network's width in order to achieve scalability. As a result, they created a wide ResNet (WRN) with the same precision. Despite the fact that WRN has more parameters than ResNet, its inference time is substantially faster. Following that, DenseNet and ResNeXt created a composite zoom version that took both depth and width into account. Enhancement at runtime is a common strategy for image pyramid reasoning. It takes an input image, scales it to various resolutions, and then feeds these pyramids into a trained CNN. In the end, the network will The final result incorporates multiple sets of outputs. The size scaling of the input image is performed by Redmon et al. [5] using the above technique. To fine-tune the trained DarkNet-53, they employed a greater input image resolution.

The field of network architecture search (NAS) has seen a lot of development in recent years, and NASFPN looks for the combined path of the feature pyramid. NAS-FPN can be thought of as a model scaling technology that is mostly used at the stage level. Efficient Net employs a compound zoom search that takes into account depth, width, and input size. EfficientNet's core design concept is to break the target detector module into several functions, then scale the picture size, width, BiFPN layer, and #box/class layer. Spine Net, which focuses on network architecture search for the overall architecture of the fish-shaped target detector, is another design that leverages the NAS principle. This design approach can finally result in a structure that is proportionally arranged. Another NAS-designed network, RegNet, fixes the number of stages and input resolution while also taking into account the depth, width, bottleneck ratio, group width, and other features of each stage in depth, initial width, slope, quantization, and group width. Finally, utilize these six factors to find the composite model's scale. The methods described above are all excellent, but there are only a few ways to examine the relationship between distinct parameters. The subsequent phase is to concern with the measureable aspects that change, such as the number of parameters with qualitative factors, after scaling the proposed target detector.

Our aim when building an effective model scale approach is that the lower/higher the quantitative cost, we wish to raise or decrease is better when the scale increases or decreases. In Section 3.1, we illustrate and examine a variety of standard CNN models in order to better understand their quantifiable costs as image size, layer count, and channel count change. ResNet, ResNext, and Darknet are the CNNs we've chosen. The size, depth, and width of the image, all increases the computing cost. Quadratic, linear, and quadratic growth are the three types of growth. The CSPNet introduced by Wang et al. [9] can be used to reduce parameters and calculations in a variety of CNN designs. It also enhances accuracy and cuts down on reasoning time. We tested it on ResNet, ResNeXt, and DarkNet and discovered a difference in calculation amount. Table 1 shows the calculation amount (flop) for Res Layer, ResX Layer, Dark Layer, and ResDark Layer.

The new design can lower ResNet, ResNeXt, and Dark-Net's calculation amount (flop) by 23.5%, 46.7%, and 50.0%, respectively, and ResDark layer's calculation amount (flop) by 72%.

Table 1 Calculation amount (flop) for Res Layer, ResX Layer, Dark Layer and ResDark Layer

Model	Original	To CSP
Res Layer	$\frac{17\ w\,h\,k\,b^2}{16}$	$w\,h\,b^2\left(\frac{3}{4}+\frac{13\ k}{16}\right)$
ResX Layer	$\frac{137\ w\,h\,k\,b^2}{128}$	$w\,h\,b^2\left(\frac{3}{4}+\frac{73\ k}{128}\right)$
Dark Layer	$5\ w\,h\,k\,b^2$	$w\,h\,b^2\left(\frac{3}{4}+\frac{5\ k}{2}\right)$
ResDark Layer	$\frac{22\ w\,h\,k\,b^2}{16}$	$w\,h\,b^2\left(\frac{3}{4}+\frac{27\ k}{8}\right)$

3.2 Scaled-YOLOv4

Scaling YOLOv4 for general GPUs, low-end GPUs was our main focus. On general-purpose GPUs, YOLOv4 is designed for real-time target identification. We redesign YOLOv4 to YOLOv4-csp in this area to acquire the optimum speed/accuracy trade-off.

Backbone: The residual block in the architecture of CSP DarkNet-53 does not include the cross-stage processing down-sampling convolution calculation. As a result, the amount of computation in each level of the CSPDarkNet is $w\ h\ b\ ^2(9/4 + 3/4 + 5\ k/2)$. The preceding formula shows that the CSPDarkNet stage has a larger computational advantage than the Darknet stage only when $k > 1$. SPDarknet53 has a total of 1-2-8-8-4 residual layers in each stage. To obtain a better speed/accuracy trade-off, we switched the first CSP stage to the original DarkNet residual layer [44].

Neck: to effectively reduce the amount of calculation in YOLOv4, we incorporate the CSP structure into the PAN design. Table 1 depicts the PAN architecture calculation list (the column of "Original"). It primarily combines features from various feature pyramids before passing through two sets of inverse DarkNet residual layers with no shortcut links. Table 2 shows the architecture of the new calculation list after cspization (the column of "To CSP"). This new update effectively cuts the number of calculations by 40% [44].

SPP: in the neck, the SPP module was initially placed in the midst of the first calculation list group. As a result, we also place the SPP module in the first CSPPAN calculation list group's middle position [44].

Finally, the inference time is used as a limitation for extra width scaling. Experiments reveal that YOLOv4-P8 can achieve real-time performance in 62 frames per second video when the width scaling factor is 1. On an edge device, the width scaling factor is equivalent to 1.25, allowing for real-time performance in 32 fps video.

Half of the output is carried via the main path (generating more semantic information with a large receptive field). The second half of the signal, on the other hand, takes a detour (retaining more spatial information with a small receptive field).

3.3 Scaled YOLOv4 lite loss function

The loss functions is calculated by using Eqs. (1) to (4):

$$bx = \sigma\left(tx\right) \times 2 - 0.5 + c_y, \tag{1}$$

$$by = \sigma\left(ty\right) \times 2 - 0.5 + c_y, \tag{2}$$

$$bw = \left(\sigma\left(ty\right) \times 2\right)^2 - 0.5 + p_w, \tag{3}$$

$$bh = \left(\sigma\left(t_h\right) \times 2\right)^2 - 0.5 + p_h. \tag{4}$$

4 Results and discussion

The suggested scaled-YOLOv4 was tested using the MSCOCO 2017 target detection dataset. The SGD optimizer is used to train the scaled-YOLOv4 models from scratch. The Google Colab server was used to train and test the suggested model. YOLOv4-tiny has 600 epochs of training, YOLOv4-CSP has 300 epochs of training, and YOLOv-CSP ResDark 154 has 300 epochs of training before using stronger data augmentation methods to train 150 epochs. We use k-means and evolutionary algorithms to calculate the hyperparameters of Lagrangian multipliers, such as anchor points, learning rate, and varying degrees of data augmentation approaches. Genetic algorithms, genetic programming, differential evolution, evolution methods, particle swarm optimization, and evolutionary programming are a few examples of the various forms of evolutionary algorithms. The GA approach is used, in which the algorithms update the parameters (called multipliers) adaptively so that the corresponding penalized function dynamically changes its optimal from the unconstrained minimum point to the constrained minimum point with iterations [43]. This avoids the need for a constant penalty parameter throughout the optimization process.

We investigated into how CSPization affected the number of parameters, the quantity of work, throughput, and average accuracy of several models. We use DarkNet-53 as the backbone for ablation experiments, as well as FPN with SPP (FPNSPP) and PAN with SPP (PANSPP). The Average Precision is calculated after CSPization of several DNN model for validation of the results.

We used the Leaky ReLU (Leaky) and Mish activation functions to compare the parameters used in the amount of calculation and throughput, respectively. The CSPized model has enhanced Batch 8 throughput and AP while reducing the amount of parameters and calculations by 32%. If you want to keep the frame rate the same after CSP, you'll need to add more layers or use a more complex activation method. The Batch 8 throughput of CD53s-CFPNSPP-Mish and CD53sCPANSPP-Leaky is the same as D53-FPNSPP-Leaky, but they perform better when computational resources are limited. Table 2 shows the results of the YOLOv4 light model analysis for different backbone, neck, and activation. Table 3 shows the ablation analysis of the training plan with and without fine-tuning of Scaled YOLO with P8.

Table 2 Analysis of YOLOv4 lite model is done for different backbone, neck and activation

Backbone	Neck	Activation	#Parameters	FLOPs	Batch 8 FPS	AP val
D53 our model	CPANSPP	Mish	48M	102B	216	52.8%
D53	FPNSSP	Leaky	63 M	142B	208	43.5%
D53	FPNSSP	Mish	63 M	142B	196	45.3%
CD53s	CFPNSSP	Leaky	43M	97B	222	45.7%
CD53s	CFPNSSP	Mish	43M	97B	208	46.3%
D53	PANSSP	Leaky	78M	160B	196	46.5%
D53	PANSSP	Mish	78M	160B	185	46.9%

Table 3 Ablation study of training schedule with and without fine-tuning of Scaled YOLO with P8

Model	Scratch	Fine tune	AP val	AP val 50	AP val 75
YOLOv4-P5	300	–	50.5%	68.9%	55.2%
YOLOv4-P5	300	150	51.2%	69.8%	56.2%
YOLOv4-P6	300	–	53.4%	71.5%	58.5%
YOLOv4-P6	300	150	53.9%	72.0%	59.0%
YOLOv4-P7	300	–	50.5%	72.4%	59.7%
YOLOv4-P7	300	150	54.6%	72.9%	–
YOLOv4-P8 our model	300	150	58.2 %	74.8 %	64.12%

All YOLOv4 models, including the YOLOv4-CSP, YOLOv4-P5, YOLOv4-P6, and YOLOv4-P7, are the finest in every metric. The inference speed of YOLOv4-CSP is 1.9 times faster than EfficientDet-D3 with the same accuracy (47.5% vs. 47.5%). When YOLOv4-P5 and EfficientDet-D5 are compared, the accuracy is the same (51.4% vs. 51.5%), and the inference speed is 2.9 times faster. For YOLOv4-P6 and EfficientDet-D7 accuracy is 54.3% vs. 53.7%, YOLOv4-P7 and EfficientDet-D7x accuracy is 55.4% vs. 55.1%, and YOLOv4-P8 and ResDark-D8x accuracy is 58.2% vs. 55.0%.

YOLOv4-P6, YOLOv4-P7, and YOLOv4-P8 are 3.7 times, 2.3 times, and 4.2 times faster, respectively, in inference speed. The comparison of our model with YOLOv3-SPP and YOLOv4-CSP is as shown in Table 4. The experimental results of the YOLOv4 large model and for modified YOLOv4 research are compared in Table 5.

Table 5 Comparison of P5, P6, P7 and scaled P8 model

Model	AP	Ap50	AP75
YOLOv4-P5 with TTA	52.5%	70.3%	58.3%
YOLOv4-P6 with TTA	58.2%	73.8%	63.3%
YOLOv4-P7 with TTA	58.2%	73.8%	63.3%
YOLOv4-P8 with TTA	58.2%	73.8%	63.3%

Fig. 2 shows plot of mAP0.5:0.95 vs. number of epochs, Fig. 3 shows plot of classification_loss vs. number of epochs, Fig. 4 shows plot of GIOU_loss vs. number of epochs and Fig. 5 shows plot of object_detection_los vs. number of epochs.

Figs. 6 and 7 shows results for testing model on testing dataset. Figs. 6 and 7 shows plot of Precision and recall vs. number of epochs.

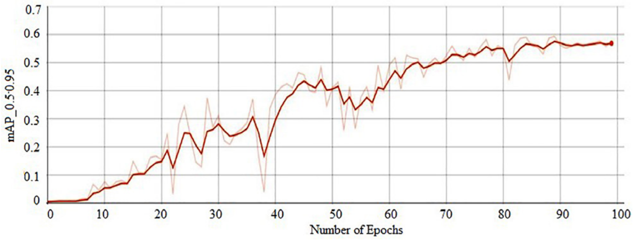

Fig. 2 Plot of mAP_0.5:0.95 vs. number of epochs

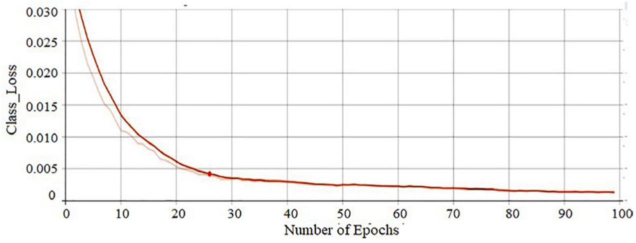

Fig. 3 Plot of classification loss vs. number of epochs

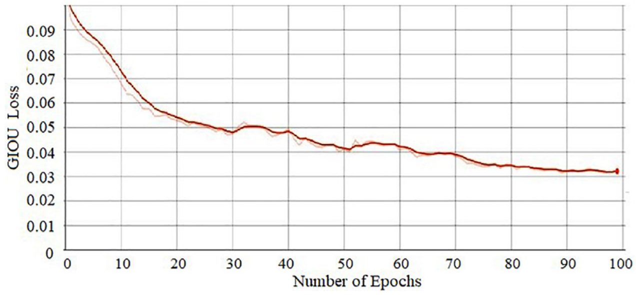

Fig. 4 Plot of GIOU loss vs. number of epochs

Table 4 Comparison of our model with YOLOv3-SPP and YOLOv4-CSP

Method	Back bone	Size	FPS	AP	AP50	AP75	APS	APM	APL
YOLOv3-SPP	D53	608	73	36.2	60.6	38.2	20.6	37.4	46.1
YOLOv4-CSP	CD53s	512	97	46.2	64.2	50.2	24.6	50.4	61.9
Ours	CD53s	512	140	50.8	63.6	52.5	28.4	51.5	62.7

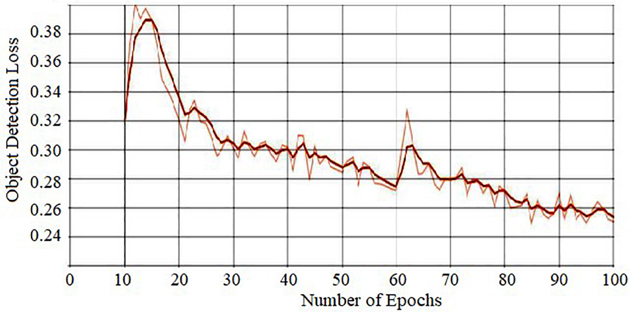

Fig. 5 Plot of object detection loss vs. number of epochs

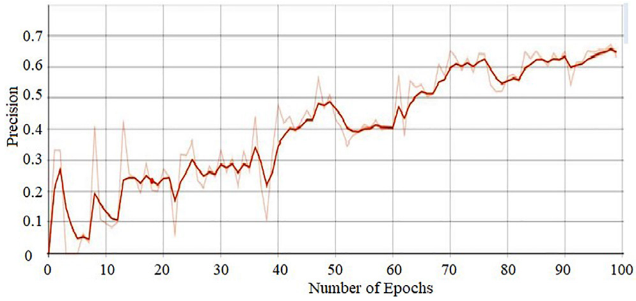

Fig. 6 Plot of precision vs. number of epochs

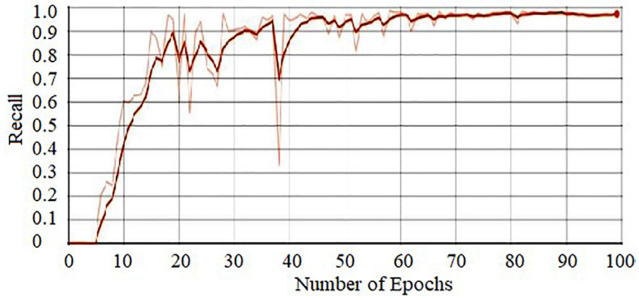

Fig. 7 Plot of recall vs. number of epochs

The analysis of proposed model for HD and Full HD video in terms of FPS and Accuracy is as shown in Table 6, which indicates the HD and full HD video can be real time for detection. The plot of various metrics vs. number of epochs is as shown in Fig. 8. The original and output images (detected images) by the proposed technique are as shown in Fig. 9.

5 Conclusion

Object detection, which combines object categorization and object location within a scene, is regarded one of the most difficult challenges in this subject of computer vision. Deep Neural Networks (DNNs) have recently been shown

Table 6 Performance analysis of proposed scaled YOLOv4 lite model when tested on real time HD and Full HD video

Video resolution	FPS	Accuracy %
1920 × 1080	28	86.09
1280 × 720	32	73.3

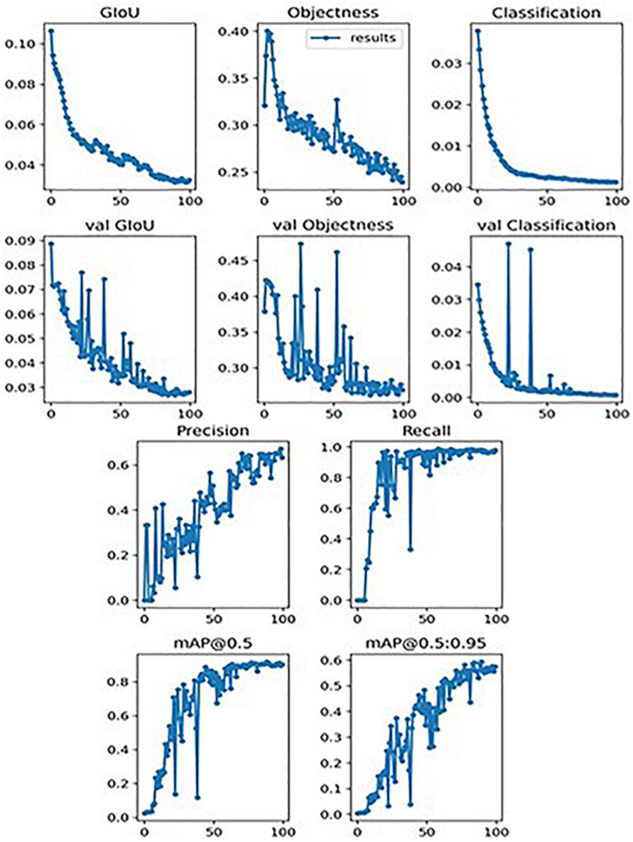

Fig. 8 Plot of various metrics vs. number of epochs

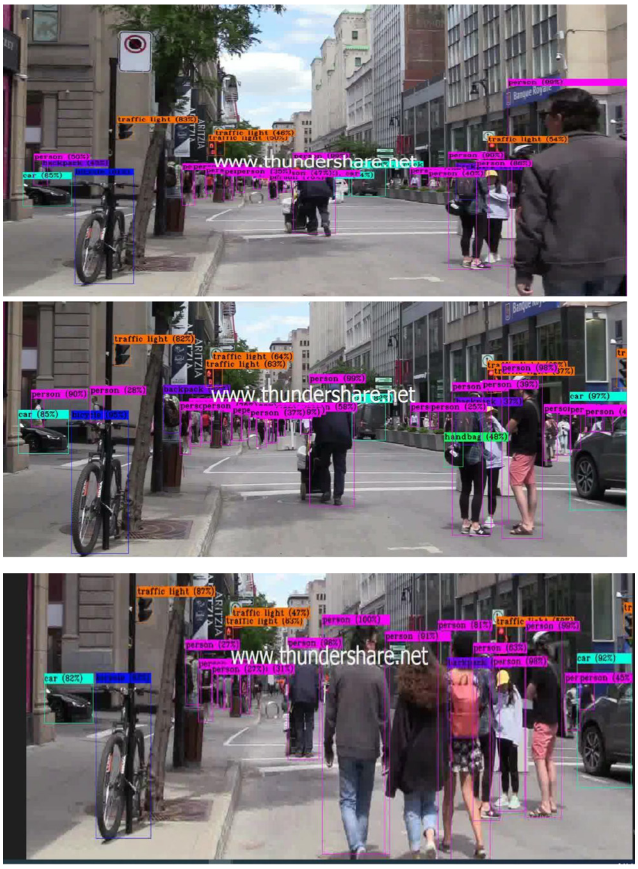

Fig. 9 Original and output images of proposed model

to perform better than other approaches in terms of object detection. To get YOLOv4-CSP-P5-P6-P7-P8 networks, we suggested the Scaled YOLOv4 that utilizes optimal network scaling strategies. When compared to ResNet-based architectures, CSPDarkNet-53 backbone has a higher accuracy in object detection while also having a superior categorization performance. The results of the experiments reveal that our YOLOv4 lite model outperforms the state-of-the-art technique. The proposed scaled model, which uses the backbone CSP DarkNet-53 and the neck CPANSPP, has 48 M parameters and 216 B flops, which is better than the current state-of-the-art. When evaluated on the testing data set, it too runs at 216 frames per second and has a mAP of 52.8%. It can run up to 28 fps and has an accuracy of 86.09% when tested on real-time video with resolutions 1920 x 1080 (full HD). AP = 50.81%, AP @50 = 63.6%, and AP @75 = 52.5% for CSPDarkNet-53 backbone.

The future scope is to increase FPS (Speed) of object detection in high-resolution video using next version YOLO model and one or two GPUs.

References

[1] Srivastava, G., Srivastava, R. "Salient object detection using background subtraction, Gabor filters, objectness and minimum directional backgroundness", Journal of Visual Communication and Image Representation, 62, pp. 330–339, 2019. https://doi.org/10.1016/j.jvcir.2019.06.005

[2] Hosaka, T., Kobayashi, T., Otsu, N. "Object Detection Using Background Subtraction and Foreground Motion Estimation", IPSJ Transactions on Computer Vision and Applications, 3, pp. 9–20, 2011. https://doi.org/10.2197/ipsjtcva.3.9

[3] Lawal, O. M. "YOLOMuskmelon: Quest for Fruit Detection Speed and Accuracy Using Deep Learning", IEEE Access, 9, pp. 15221–15227, 2021. https://doi.org/10.1109/ACCESS.2021.3053167

[4] Pestana, D., Miranda, P. R., Lopes, J. D., Duarte, R. P., Véstias, M. P., Neto, H. C., De Sousa, J. T. "A Full Featured Configurable Accelerator for Object Detection With YOLO", IEEE Access, 9, pp. 75864–75877, 2021. https://doi.org/10.1109/access.2021.3081818

[5] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. "You Only Look Once: Unified, Real-Time Object Detection", In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788. https://doi.org/10.1109/cvpr.2016.91

[6] Zhu, H., Wei, H., Li, B., Yuan, X., Kehtarnavaz, N. "Real-Time Moving Object Detection in High-Resolution Video Sensing", Sensors, 20(12), 3591, 2020. https://doi.org/10.3390/s20123591

[7] HariPriya, D., Aman, S. "Moving object detection and classification using background subtraction algorithm and mixture of Gaussian algorithm", International Journal of Advanced Science and Technology, 29(7), pp. 2979–2988, 2020. [online] Available at: http://sersc.org/journals/index.php/IJAST/article/view/18209 [Accessed: 01 November 2022]

[8] Kim, S., Kim, H. "Zero-Centered Fixed-Point Quantization With Iterative Retraining for Deep Convolutional Neural Network-Based Object Detectors", IEEE Access, 9, pp. 20828–20839, 2021. https://doi.org/10.1109/access.2021.3054879

[9] Wang, X., Song, J. "ICIoU: Improved Loss Based on Complete Intersection Over Union for Bounding Box Regression", IEEE Access, 9, pp. 105686–105695, 2021. https://doi.org/10.1109/access.2021.3100414

[10] Sambolek, S., Ivasic-Kos, M. "Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors", IEEE Access, 9, pp. 37905–37922, 2021. https://doi.org/10.1109/access.2021.3063681

[11] Bhatti, M. T., Khan, M. G., Aslam, M., Fiaz, M. J. "Weapon Detection in Real-Time CCTV Videos Using Deep Learning", IEEE Access, 9, pp. 34366–34382, 2021. https://doi.org/10.1109/access.2021.3059170

[12] Maddalena, L., Petrosino, A. "Background Subtraction for Moving Object Detection in RGBD Data: A Survey", Journal of Imaging, 4(5), 71, 2018. https://doi.org/10.3390/jimaging4050071

[13] Shaikh, S. H., Saeed, K., Chaki, N. "Moving Object Detection Using Background Subtraction", In: Moving Object Detection Using Background Subtraction, Springer, 2014, pp. 15–23. ISBN 978-3-319-07385-9 https://doi.org/10.1007/978-3-319-07386-6_3

[14] Zheng, Z., Zhao, J., Li, Y. "Research on Detecting Bearing-Cover Defects Based on Improved YOLOv3", IEEE Access, 9, pp. 10304–10315, 2021. https://doi.org/10.1109/access.2021.3050484

[15] Wang, Z.-Z., Xie, K., Zhang, X.-Y., Chen, H.-Q., Wen, C., He, J.-B. "Small-Object Detection Based on YOLO and Dense Block via Image Super-Resolution", IEEE Access, 9, pp. 56416–56429, 2021. https://doi.org/10.1109/access.2021.3072211

[16] Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., Pattabiraman, V. "Comparative analysis of deep learning image detection algorithms", Journal of Big Data, 8(1), 66, 2021. https://doi.org/10.1186/s40537-021-00434-w

[17] Asif, A., Khatoon, S., Hasan, M. M., Alshamari, M. A., Abdou, S., Elsayed, K. M., Rashwan, M. "Automatic analysis of social media images to identify disaster type and infer appropriate emergency response", Journal of Big Data, 8(1), 83, 2021. https://doi.org/10.1186/s40537-021-00471-5

[18] Knausgård, K. M., Wiklund, A., Sørdalen, T. K., Halvorsen, K. T., Kleiven, A. R., Jiao, L., Goodwin, M. "Temperate fish detection and classification: a deep learning based approach", Applied Intelligence, 52(6), pp. 6988–7001, 2021. https://doi.org/10.1007/s10489-020-02154-9

[19] Saponara, S., Elhanashi, A., Gagliardi, A. "Real-time video fire/smoke detection based on CNN in antifire surveillance systems", Journal of Real-Time Image Processing, 18(3), pp. 889–900, 2021. https://doi.org/10.1007/s11554-020-01044-0

[20] Kumar, S., Yadav, D., Gupta, H., Verma, O. P., Ansari, I. A., Ahn, C. W. "A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management", Electronics, 10(1), 14, 2021. https://doi.org/10.3390/electronics10010014

[21] Lawal, M. O. "Tomato detection based on modified YOLOv3 framework", Scientific Reports, 11(1), 1447, 2021. https://doi.org/10.1038/s41598-021-81216-5

[22] Yu, J., Zhang, W. "Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4", Sensors, 21(9), 3263, 2021. https://doi.org/10.3390/s21093263

[23] Ismail, A., Mehri, M., Sahbani, A., Ben Amara, N. "Performance Benchmarking of YOLO Architectures for Vehicle License Plate Detection from Real-time Videos Captured by a Mobile Robot", In: Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Vienna, Austria, 2021, pp. 661–668. ISBN 978-989-758-488-6 https://doi.org/10.5220/0010349106610668

[24] Wu, L., Ma, J., Zhao, Y., Liu, H. "Apple Detection in Complex Scene Using the Improved YOLOv4 Model", Agronomy, 11(3), 476, 2021. https://doi.org/10.3390/agronomy11030476

[25] Schütz, A. K., Schöler, V., Krause, E. T., Fischer, M., Müller, T., Freuling, C. M., Conraths, F. J., Stanke, M., Homeier-Bachmann, T., Lentz, H. H. K. "Application of YOLOv4 for Detection and Motion Monitoring of Red Foxes", Animals, 11(6), 1723, 2021. https://doi.org/10.3390/ani11061723

[26] Lee, C.-H., Lin, C.-W. "A Two-Phase Fashion Apparel Detection Method Based on YOLOv4", Applied Sciences, 11(9), 3782, 2021. https://doi.org/10.3390/app11093782

[27] Janardhan, D., Madhuri, B. J., Kumar, H., Sahu, K., Suhas, S. "Object Detection with Voice Feedback using YOLOv4", International Journal of Engineering Science and Computing, 11(6), pp. 28361–28363, 2021. [online] Available at: https://ijesc.org/upload/dcaa8478943be4ea59f3a6189de13f91.Object%20Detection%20with%20Voice%20Feedback%20using%20YOLOv4%20(1).pdf [Accessed: 08 August 2021]

[28] Růžička, V., Franchetti, F. "Fast and accurate object detection in high resolution 4K and 8K video using GPUs", In: 2018 IEEE High Performance extreme Computing Conference (HPEC), Waltham, MA, USA, 2018, pp. 1–7. ISBN 978-1-5386-5990-8 https://doi.org/10.1109/hpec.2018.8547574

[29] Ammar, A., Koubaa, A., Ahmed, M., Saad, A., Benjdira, B. "Vehicle Detection from Aerial Images Using Deep Learning: A Comparative Study", Electronics, 10(7), 820, 2021. https://doi.org/10.3390/electronics10070820

[30] Cao, Z., Shao, M., Xu, L., Mu, S., Qu, H. "MaskHunter: real-time object detection of face masks during the COVID-19 pandemic", IET Image Processing, 14(16), pp. 4359–4367, 2020. https://doi.org/10.1049/iet-ipr.2020.1119

[31] Liu, T., Pang, B., Zhang, L., Yang, W., Sun, X. "Sea Surface Object Detection Algorithm Based on YOLO v4 Fused with Reverse Depthwise Separable Convolution (RDSC) for USV", Journal of Marine Science and Engineering, 9(7), 753, 2021. https://doi.org/10.3390/jmse9070753

[32] Ma, L., Chen, Y., Zhang, J. "Vehicle and Pedestrian Detection Based on Improved YOLOv4-tiny Model", Journal of Physics: Conference Series, 1920, 012034, 2021. https://doi.org/10.1088/1742-6596/1920/1/012034

[33] Liu, H., Fan, K., Ouyang, O., Li, N. "Real-Time Small Drones Detection Based on Pruned YOLOv4", Sensors, 21(10), 3374, 2021. https://doi.org/10.3390/s21103374

[34] Magalhães, S. A., Castro, L., Moreira, G., dos Santos, M. N., Cunha, M., Dias, J., Moreira, A. P. "Evaluating the Single-Shot MultiBox Detector and YOLO Deep Learning Models for the Detection of Tomatoes in a Greenhouse", Sensors, 21(10), 3569, 2021. https://doi.org/10.3390/s21103569

[35] Mahto, P., Garg, P., Seth, P., Panda, J. "Refining Yolov4 for Vehicle Detection", International Journal of Advanced Research in Engineering and Technology (IJARET), 11(5), pp. 409–419, 2020. [online] Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3628439 [Accessed: 01 November 2022]

[36] Bochkovskiy, A., Wang, C.-H., Liao, H.-Y. M. "YOLOv4: Optimal Speed and Accuracy of Object Detection", [cs.CV], arXiv:2004.10934v1, Cornell University, Ithaca, NY, USA, 2020 https://doi.org/10.48550/arXiv.2004.10934

[37] Kim, C., Lee, J., Han, T., Kim, Y.-M. "A hybrid framework combining background subtraction and deep neural networks for rapid person detection", Journal of Big Data, 5(1), 22, 2018. https://doi.org/10.1186/s40537-018-0131-x

[38] Yu, Z., Shen, S., Shen, C. "A real-time detection approach for bridge cracks based on YOLOv4-FPM", Automation in Construction, 122, 103514, 2021. https://doi.org/10.1016/j.autcon.2020.103514

[39] Zhang, D., Chen, X., Ren, Y., Xu, N., Zheng, S. "Smart-YOLO: A Light-Weight Real-time Object Detection Network", Journal of Physics: Conference Series, 1757, 012096, 2021. https://doi.org/10.1088/1742-6596/1757/1/012096

[40] Bohush, R., Ablameyko, S., Ihnatsyevaa, S., Adamovskiy, Y. "Object detection algorithm for high resolution images based on convolutional neural network and multiscale processing", presented at CMIS-2021: The Fourth International Workshop on Computer Modeling and Intelligent Systems, Zaporizhzhia, Ukraine, Apr. 27, 2021. https://doi.org/10.32782/cmis/2864-12

[41] Kadadi, R., Devpriya, G., Taifa, I. W. R. "Deploying Background Subtraction Approach for Tracking and Detecting Moving Objects", Proceedings on Engineering Sciences, 2(2), pp. 127–136, 2020. https://doi.org/10.24874/pes02.02.003