ISSN: 0067-2904

Improving the YOLOV7 Algorithm for Object Detection within Recorded Videos

Asmaa Hasan Alrubaie [1*], Maisa'a Abid Ali Khodhe[2], Ahmed Talib Abdulameer [3]
[1] Department of Computer Sciences, University of Technology, Baghdad, Iraq
[2] Department of Computer Engineering, University of Technology, Baghdad, Iraq
[3] Department of Information Technology, Middle Technical University, Baghdad, Iraq

Received: 4/11/2022 Accepted: 14/3/2023 Published: 29/2/2024

.**Abstract**

Object detection algorithms play an important role in detecting people in surveillance videos. In recent years, with the rapid development of deep learning, the performance of object detection has improved by leaps and bounds, and the scheme of object detection by the YOLOV7 algorithm has also been born. Traditional object detection methods often fail to achieve a balance between speed and accuracy. To address these issues, in this research, an improved YOLOv7 algorithm performance is proposed to get the best speed-to-accuracy balance compared to state-of-the-art object detection within recorded videos using an effective compression method. This method calculates the difference between frames of video, and by using the zero difference approach by removing the duplicate frames from the recorded video and choosing only the meaningful frames based on many variables, including frame size, frame details, and the distance of the frames, influence the choice of a meaningful frame, and this will reduce the size of the video by eliminating the frames comparable to those chosen.

Additionally, any other datasets or pre-trained weights have not been used; YOLOv7 has been exclusively trained on the MS COCO dataset from scratch. In order to ensure the effectiveness of this approach, numerous detection systems are used in this work. Additionally, positive performance results to reduce the processing time required for object detection have been attained.

Keywords: Object Detection, YOLOv7, remove duplicate frames algorithm, MS COCO dataset.

تحسين خوارزمية YOLOv7 لإكتشاف الكائنات ضمن الفيديو المسجل

أسماء حسن الربيعي [1*] , ميساء عبد علي خضر [2], احمد طالب عبد الامير [3]
[1]قسم علوم الحاسوب, الجامعة التكنولوجية, بغداد, العراق
[2] قسم هندسة الحاسوب, الجامعة التكنولوجية, بغداد, العراق
[3]قسم تكنولوجيا المعلومات, الجامعة التقنية الوسطى, بغداد, العراق

*Email: cs.20.41@grad.uotechnology.edu.iq

الخلاصة

تلعب خوارزميات اكتشاف الكائنات دورًا مهمًا في اكتشاف الأشخاص داخل مقاطع فيديو المراقبة. في السنوات الأخيرة ، مع التطور السريع للتعلم العميق ، تم تحسين أداء الكشف عن الكائنات على قدم وساق ، غالباً ما تفشل الطرق التقليدية للكشف عن الكائنات YOLOV7.وبدأ يولّد مفهوم اكتشاف الكائنات بواسطة في تحقيق التوازن بين السرعة والدقة في الكشف عن الكائنات. لمعالجة هذه المشكلة ، نقترح في هذا البحث للحصول على افضل توازن بين السرعة والاداء للكشف عن YOLOV7 تحسين اداء خوارزمية الكائنات داخل مقاطع الفيديو المسجلة باستخدام طريقة ضغط فعالة تعتمد على حساب الفرق بين اطارات الفيديو، وباستخدام نهج الفرق الصفري يتم إزالة الإطارات المكررة من الفيديو المسجل واختيار الإطارات ذات المعنى فقط , بناءً على العديد من المتغيرات ، بما في ذلك حجم الإطار ، تؤثر تفاصيل الإطار ومسافة الإطارات على اختيار الإطار ذي المعنى ، وسيؤدي ذلك إلى تقليل حجم الفيديو عن طريق إزالة الإطارات المماثلة لتلك المختارة.

YOLOv7 بالإضافة إلى ذلك ، لا نستخدم أي مجموعات بيانات أخرى أو أوزان مُدربة مسبقًا ؛ تم تدريب , يتم استعمال من البداية. من أجل ضمان فعالية هذا النهج MS COCO حصريًا على مجموعة بيانات العديد من أنظمة الكشف في هذا العمل. بالإضافة إلى ذلك ، تم تحقيق نتائج أداء إيجابية لتقليل وقت المعالجة المطلوب لاكتشاف الكائن.

1. Introduction

Object detection seeks to detect particular targets in video or image sequences and determine their size, relations to other objects, location, and class. It falls under the umbrella of digital image processing, machine vision, and computer vision [1], [2], [3], and [4]. In industrial defect detection, face recognition, vehicle and traffic detection, pedestrian detection and counting, and other areas, it has often had a high research and application value [5], [6], [7], [8], [9], and [10]. Deep learning (DL) architectures depending on convolutional neural networks (CNNs) recently significantly improved object detection problems. If we merely look at accuracy results, such models have been efficiently applied to benchmark datasets for various applications, such as the MS COCO dataset [11], [12], and [13].

On the other hand, they have limits when used in real-time applications like autonomous driving. In this case, inference speed and the necessary computational resources are also crucial in addition to detection accuracy. It is challenging for practitioners to choose which architecture is best for this specific problem, given the variety of existing ones [14], [15], [16], [17], [18]. At a high level, current DL architectures for object detection using remote sensing camera data exhibit a common method: a convolutional backbone as a feature extractor and a sliding window prediction with a mixed regression and classification objective [19], [20].

Our suggested approach enables a unified comparison across several object detection systems that used Yolov7 to find objects with various configurations after removing duplicate frames from recorded videos in the surveillance system's database to speed up object detection. Thus, we use multiple systems with various settings to present a complete view of the accuracy/speed trade-off. By deleting duplicate frames from the video, this study provided an approach for speeding up detection inside recorded videos using Yolov7 while maintaining algorithm accuracy. The remaining sections of the present work are structured as follows: The second section provides a summary of the related works on object detection, including problems, major difficulties, and advancements. The third section introduces YOLOv7's structure. The performance comparison of different detection methods is covered in the fourth section. The fifth section proposes a technique. The experimental findings are described in the

sixth section, along with a description of object detection technology. Future works and this study's conclusion are covered in the seventh section.

2. Related Work

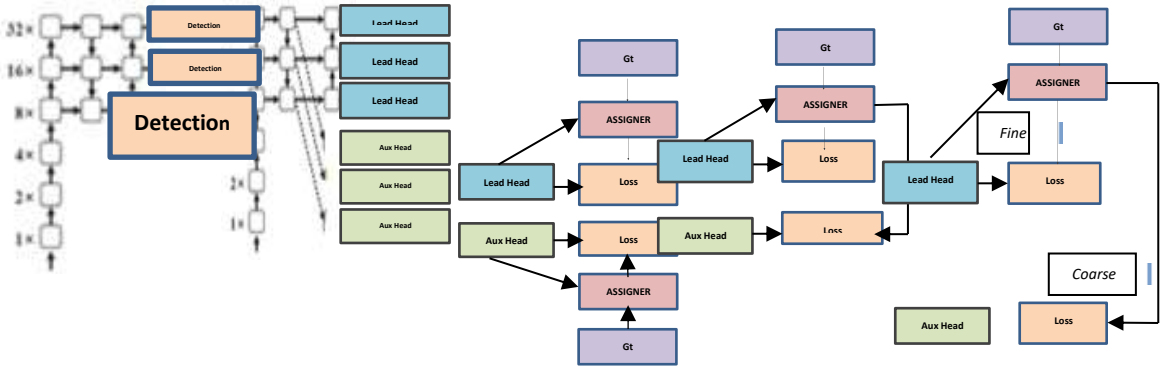
The current literature concerning object detection using the YOLOv7 algorithm suggests a shortage of active research in this area; only a few publications use the YOLOv7 algorithm for object detection. Hence, we broadened our literature review to focus on reducing the amount of data and improving object detection using the YOLOv7 algorithm.

In [2021], Christian Salim et al. presented that the demand for intelligent agriculture has grown recently. For monitoring, for example, surveillance systems are set up to assist them, particularly in remote locations. A wireless video sensor network (WVSN) could be a low-cost and energy-efficient alternative. The mechanism at WVSN typically runs periodically, sending all captured frames to the station. Wireless video sensor nodes are energy-constrained nodes that transmit frames and videos to the station while monitoring a particular region of interest based on their fields of view (FOVs). Sending this many frames to the station uses even more energy than shooting and collecting the frames for a video. Utilizing the overlap between FoVs and node-embedded lightweight algorithms, we seek to decrease the quantity of data provided to the station to contend with nodes' restricted processing and energy resources. This will lower the consumption of energy at the level of the sensor node [21].

In [2020], Mate Kristo and Marina Ivasic-Kos proposed using thermal cameras for automatic person detection within recorded video surveillance systems due to their ability to be used at night and in weather conditions where RGB cameras do not perform well. Convolutional Neural Network models originally intended for automatic person detection in RGB thermal images are compared with the performance of the standard state-of-the-art object detectors such as Faster R-CNN, SSD, Cascade R-CNN, and YOLO that were retrained on a dataset of thermal images extracted from videos that simulate illegal movements around the border and in protected areas. Videos are recorded at night in clear weather, rain, and fog, at different ranges, and with different movement types. Yolo was significantly faster than other detectors while achieving performance comparable to the best, so it was used in further experiments [22].

3. Object Detection Using YOLO v7 Algorithm

The next significant advancement in architecture search is referred to as YOLO v7. YOLO v7, a new YOLO model, is anticipated to become the next industry standard for object detection [23, 24]. The number of computations, parameters, and computational density of a model are the main considerations in constructing an effective YOLO v7 architecture. Researchers from YOLOv7 examined how re-parameterized convolution must be coupled with various networks using the gradient flow propagation paths. In the YOLOv7 architecture, the lead head is responsible for producing the output, and the auxiliary head is responsible for assisting in training. To create coarse-to-fine hierarchical labels for the auxiliary head and the lead head, respectively, YOLOv7 utilizes the prediction of the lead head as guidance. Figure 1 depicts the two suggested deep supervision label assignment mechanisms.



(a) **(b)** **(c)** **(d)** **(e)**

Figure 1: Coarse for the auxiliary and fine for the lead head label assigner. Compared to the normal model (a), the schema in (b) has an auxiliary head. Different from a typical independent label assigner (c), we proposed (d) lead head guided label assigner and (e) coarse-to-fine lead head guided label assigner [23].

The MS COCO dataset is used to assess the model [25, 26]. Over 200,000 images, 250,000 human instances, and 17 key points are included. The 57K image train2017 collection is larger than the 5K and 20K image val2017 and test-dev2017 sets combined. The train 2017 set has been used to train the model, and thr test-dev2017 and val2017 sets are used to report results. For 300 epochs, the model is trained. First, we maintain the aspect ratio while resizing the large side of the input frames to the required size. To create a square image, padding is placed on the lowest side of the frame. This guarantees that the frame size of all input videos is the same.

4. Performance Comparison OF Various Detection Systems

Object detection is currently a rather well-liked field, from conventional to deep learning techniques. In this study, object detection by the YOLOv7 algorithm is executed on recorded videos using different detection systems. The working principles of each detection system are explained in depth, and the differences and efficiencies between them are examined for each system, as listed in Table 1.

Table 1: Object Detection by YOLOv7 Algorithm in Different Detection Systems

Detection Systems	Architecture	Advantage	Disadvantage	Time processing
1. Colaboratory	The Colaboratory program works on the internet, so to execute the YOLOv7 algorithm by the Colaboratory program, one must upload the YOLOv7 algorithm to Google Drive, call it inside the Colaboratory program to execute it, and store the result again inside the drive.	1- Colaboratory is a free environment that runs entirely in the cloud, meaning you can access your project from any place, device, and time. 2- Colaboratory supports GPU, and it is free. The colaborative environment provides an additional attractive characteristic that Google offers to developers: GPU use.	1- Colaboratory runs entirely on the internet, so the execution efficiency of the YOLOv7 algorithm for object detection depends on the speed of the internet; downtime is usually cited as one of the biggest drawbacks of the Colaboratory. 2- Storing data and significant files on external service providers always poses risks, particularly when managing sensitive data.	The time required to process the YOLOv7 algorithm depends on the speed of the internet; execution of YOLOv7 by Colaboratory took about 30 minutes.
2.Anaconda3 Prompt	Anaconda represents the most popular platform for data science for IT professionals and data scientists worldwide. It represents a powerful package manager and environment manager, which are used with the command-line commands at Anaconda Prompt for Windows or in the terminal window for Linux or macOS.	1-It's open-source and free and simplifies the packages' management and deployment. It represents the standard platform for Python data science and open-source ML. 2- The download and installation of Anaconda3 Prompt from the internet to your device is easy.	1. In Anaconda3 prompt, manually install most of the packages and libraries when needed to execute the YOLOv7 algorithm. 2- Graphics Processing Unit (GPU) tends to increase the system's performance to a higher level. In general, graphics cards are expensive, depending on their model. The higher the price, the greater its performance will be. Even some laptops with dedicated graphics cards are more expensive than integrated graphics.	15 minute.
3.PyCharm 2022	PyCharm is available as a cross-platform application that is compatible with macOS, Windows, and Linux platforms. Gracefully sitting amongst the best Python IDEs, PyCharm supports Python 2 (2.7) and Python 3 (3.5 and higher) versions. PyCharm comes with a wide range of packages, modules, and tools to hasten the development of Python, cutting down on the efforts needed to do the same to a higher degree simultaneously.	1- Pycharm can be described as the industry standard for developing, training, and testing a single device. 2- The program contains graphical interfaces that make the implementation of any code easier. 3- Downloading and installing PyCharm 2022 from the internet to your device is easy, with tools like Anaconda3 Prompt.	In Colaboratory, you do not need to manually install most packages and libraries; just import them directly by calling them. In PyCharm, 2022 is a normal IDE, so you must install the libraries in a manual way.	5 minutes.

5. Proposed Method

YOLOv7 surpassed all known object detector types in speed and accuracy in a range from 5 FPS to 160 FPS and has the maximum accuracy of 56.80% AP test and 56.80% AP min-val amongst all of the known real-time object detectors with 30 FPS or higher on GPU V 100. But when executing yolov-7 on recorded video from DB of the surveillance system by different detection systems, notice it takes a long time if the video size is large. The proposed method aims to represent an efficient method to reduce the processing time of the YOLOv7 algorithm

required for object detection within recorded videos by applying the remove duplicate frames algorithm that consists of many steps, as shown below, that apply zero-difference approaches. Many frame differences between the consecutive frames are 0; in this case, those frames will be eliminated. It is better to eliminate the frames where the frame difference is equal to 0, which will minimize the number of frames.

YOLOv7 surpassed all known object detector types in speed and accuracy in a range from 5 FPS to 160 FPS and has the maximum accuracy of 56.80% AP test and 56.80% AP min-val amongst all of the known real-time object detectors with 30 FPS or higher on GPU V 100. But when executing yolov-7 on recorded video from DB of the surveillance system by different detection systems, notice it takes a long time if the video size is large. The proposed method aims to represent an efficient method to reduce the processing time of the YOLOv7 algorithm required for object detection within recorded videos by applying the remove duplicate frames algorithm that consists of many steps, as shown below, that apply zero-difference approaches. Many frame differences between the consecutive frames are 0; in this case, those frames will be eliminated. It is better to eliminate the frames where the frame difference is equal to 0, which will minimize the number of frames.

Algorithm1: Remove duplicate frames.

Inputs: Digital video.
Outputs: Reduce digital video size.

Step1: Start.
Step 2: Read the video, file name= name of the video file and path=location of the video file.
Step3: Extracting the frames from the video and storing them in an array of one dimension is [N]. // N= frame No.
Step 4: Read frames of video by utilizing the for loops

```
For i=0 to N
if frame[i] = frame[i+1]: // this condition to compare between two adjacent frames by applied
Zero Difference Approach
remove (frame)
print ("similar")

Else:
print ("non-similar")
i = i+1// i is counter to continue until all duplicate video frames are removed.
```

Step5: Now every frame of video is a unique frame of the video capture.
Step6: Convert the sequence of frames into digital video.
Step7: End.

The proposed method will be executed first by applying the remove duplicate frames algorithm to meaningful frames, in which similar frames are eliminated to reduce the size of the video. It will execute the YOLOv7 algorithm, and the result is object detection by the bounding box surrounding the object, as shown in Figure 2.

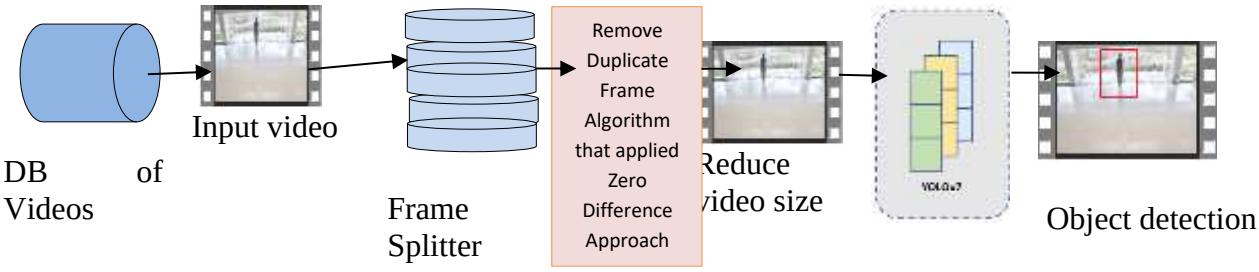


Figure 2: Structural of the proposed method.

6. Experimental Result and Discussion

To investigate the capabilities of our technique, three detection systems were chosen as subjects: Colaboratory, Anaconda3 Prompt, and Pycharm. When the proposed method was applied to surveillance videos from a static camera with a resolution of (1280 x 720) and a frame rate of 30 frames per second, all three detection systems performed better. Digital video is made up of a stream of images that are captured at regular intervals of time that are called frames. This frame is duplicated during the video. So in the proposed method, I'd like to delete the duplicated frames after applying zero difference approaches and end up with my video running at the true speed of 25 fps to reduce the processing time required for object detection by detection systems when applying the YOLOv7 algorithm, as shown in Table 2.

Table 2: Results of the processing time before and after applying the proposed method.

Detection systems	The processing time before applied proposed method	The processing time after applied proposed method
Colaboratory	30 minute	16.7 minute
Anaconda3 Prompt	15 minute	8 minutes
pycharm	5 minutes	2.8 minutes

Notice the results in Table 2: the time spent processing the same video is reduced by a different amount for each detection system, as shown in Figure 3. Colaboratory gives maximum time processing, and PyCharm gives minimum time processing; the result of Anaconda3 Prompt is the difference between them, which depends on the features of each detection system, as explained in Table 1.

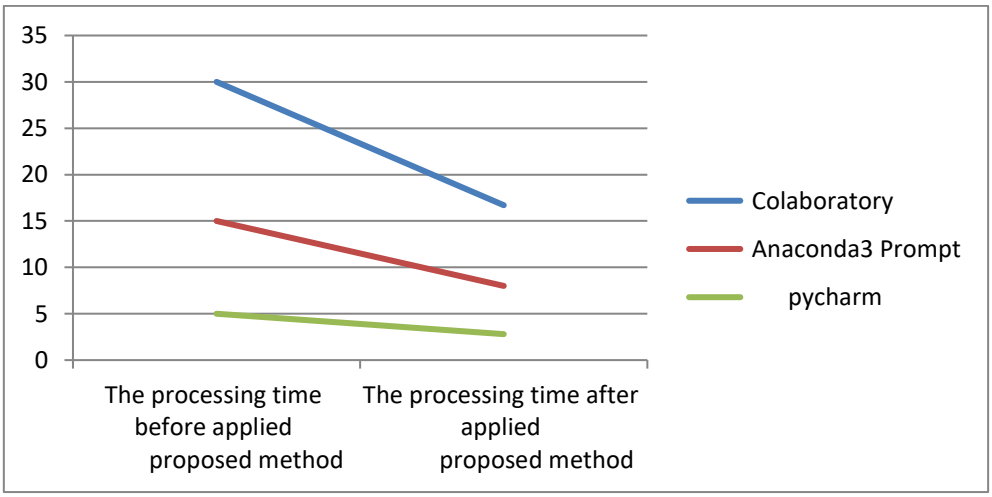


Figure 3: The processing time results before and after applying the proposed approach.

A special, straightforward questionnaire was utilized to measure the dimension of overall video quality using a subjective video quality evaluation, as shown in Table 3. After presenting the findings of a suggested method, examples of questions have been raised. The outcomes of the video quality evaluation are shown in Table 2 below. It is evident from the results that video quality improved as frame rates rose. By comparing the assessments of 25 frames to the remaining numbers of frames per second (10, 15, and 20), the maximum video quality scores are obtained.

Table 3: Subjective video quality evaluation.

Factors of the Video Quality	10 frames	15 frames	20 frames	25 frames
Rating of video colors	3.3	3.9	4.0	4.5
Rating of video borders	3.9	4.1	4.20	4.60
Rating of video contrast	3.8	4.0	4.30	4.7
flicker in the sequence of the video	Annoying	not annoying	not annoying	not annoying
Rating of continuity of movement	3.7	3.8	4.1	4.40
Smearing in the sequence of the video	Annoying	not annoying	not annoying	not annoying

7. Conclusions

This study presented a data reduction method for improving the YOLOV-7 algorithm for recorded video. The remove duplicate frames algorithm and YOLOv7 complement this approach's two complementing algorithms. The remove duplicate frames algorithm is the first step in the method, which suggests that the true frame rate of the video should be 25 frames per second by identifying similarities between successively captured frames and deciding to eliminate the frames that share those similarities so that each frame of video is a unique frame. The method's second step is object detection using the Yolov7, which produces object detection through the bounding box surrounding the object. The outcomes demonstrate that no incident in the recorded video sequence was missed after applying the suggested algorithms.

The results of the case study were evaluated by measuring different attributes of video with three detection systems, which were chosen as subjects for this purpose. Comparing the size of the video before and after applying the proposed approach, it was noticed that the number of frames became 25 fps instead of 30 fps, reducing the size of the video. As a suggestion for future work, first of all, we aim to compare the relevance of the suggested approach by comparing it with other approaches to increase the reduction of the frame to save the resolution of the video and also improve the performance of the YOLOv7 algorithm to detect an object within recorded videos in a faster way.

References

[1] L. He *et al.*, "End-to-end video object detection with spatial-temporal transformers," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1507–1516.

[2] Y. Liu, L. Geng, W. Zhang, Y. Gong, and Z. Xu, "Survey of video based small target detection," *J. Image Graph.*, vol. 9, no. 4, pp. 122–134, 2021.

[3] A. M. Rekavandi, L. Xu, F. Boussaid, A.-K. Seghouane, S. Hoefs, and M. Bennamoun, "A Guide to Image and Video based Small Object Detection using Deep Learning: Case Study of Maritime Surveillance," *arXiv Prepr. arXiv2207.12926*, 2022.

[4] A. Panda and S. M. Muniz, "Smart home with neural network based object detection," *Big Data Comput. Visions*, vol. 2, no. 1, pp. 40–48, 2022.

[5] H. Liu, F. Sun, J. Gu, and L. Deng, "SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode," *Sensors*, vol. 22, no. 15, p. 5817, 2022.

[6] A. K. Maisa'a and A.-M. S. Rahma, "Suggesting an Analysis System for Monitoring Free hand Gymnastics for Training Youth," in *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, 2019, pp. 162–166.

[7] A. M. S. Rahma and K. Maisa'a Abid Ali, "Proposing an analysis system to monitoring weightlifting based on training (snatch and clean and jerk)," *Baghdad Sci. J.*, vol. 15, no. 4, 2018.

[8] H.-C. Shih, "A survey of content-aware video analysis for sports," *IEEE Trans. circuits Syst. video Technol.*, vol. 28, no. 5, pp. 1212–1231, 2017.

[9] L. Lin, Y. Xu, X. Liang, and J. Lai, "Complex background subtraction by pursuing dynamic spatio-temporal models," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3191–3202, 2014.

[10] A. M. S. Rahma, M. A. S. Rahma, and M. A. S. Rahma, "Automated analysis for basketball free throw," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2015, pp. 447–453.

[11] T. Diwan, G. Anirudh, and J. V Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, pp. 1–33, 2022.

[12] X. Du, X. Wang, G. Gozum, and Y. Li, "Unknown-Aware Object Detection: Learning What You Don't Know from Videos in the Wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13678–13688.

[13] J. Yin, J. Shen, X. Gao, D. Crandall, and R. Yang, "Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[14] H. Zhang, Y. Wang, Y. Liu, and N. Xiong, "IFD: An Intelligent Fast Detection for Real-Time Image Information in Industrial IoT," *Appl. Sci.*, vol. 12, no. 15, p. 7847, 2022.

[15] J. Yang, S. Liu, Z. Li, X. Li, and J. Sun, "Real-time Object Detection for Streaming Perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5385–5395.

[16] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," *Intell. Comput. Image Process. Based Appl.*, pp. 1–16, 2020.

[17] Z. Xie, Y. Song, J. Wu, Z. Li, C. Song, and Z. Xu, "MDS-Net: A Multi-scale Depth Stratification Based Monocular 3D Object Detection Algorithm," *arXiv Prepr. arXiv2201.04341*, 2022.

[18] M. Terreran, A. G. Tramontano, J. C. Lock, S. Ghidoni, and N. Bellotto, "Real-time object detection using deep learning for helping people with visual impairments," in *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)*, 2020, pp. 89–95.

[19] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sens.*, vol. 13, no. 1, p. 89, 2020.

[20] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-Maadeed, "Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches," *arXiv Prepr. arXiv2206.08016*, 2022.

[21] C. Salim and N. Mitton, "Image similarity based data reduction technique in wireless video sensor networks for smart agriculture," in *International Conference on Advanced Information Networking and Applications*, 2021, pp. 448–459.

[22] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using YOLO," *IEEE access*, vol. 8, pp. 125459–125476, 2020.

[23] K. Jiang *et al.*, "An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation," *Agriculture*, vol. 12, no. 10, p. 1659, 2022.

[24] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and T. Alsboui, "Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections," *Sensors*, vol. 22, no. 18, p. 6927, 2022.

[25] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, 2014, pp. 740–755.

[26] D. K. Sharma, "Information Measure Computation and its Impact in MI COCO Dataset," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, vol. 1, pp. 1964–1969.