Journal of Engineering and Applied Science

RESEARCH Open Access

Performance analysis of deep learning-based object detection algorithms on COCO benchmark: a comparative study

Jiya Tian[1], Qiangshan Jin[1], Yizong Wang[1], Jie Yang[1], Shuping Zhang[2*] and Dengxun Sun[1]

*Correspondence:
zsp2714105538@163.com

[1] School of Information Engineering, Xinjiang Institute of Technology, Aksu, Xinjiang 843100, China
[2] Scientific Research Department, Xinjiang Institute of Technology, Aksu, Xinjiang 843100, China

Abstract

This paper thoroughly explores the role of object detection in smart cities, specifically focusing on advancements in deep learning-based methods. Deep learning models gain popularity for their autonomous feature learning, surpassing traditional approaches. Despite progress, challenges remain, such as achieving high accuracy in urban scenes and meeting real-time requirements. The study aims to contribute by analyzing state-of-the-art deep learning algorithms, identifying accurate models for smart cities, and evaluating real-time performance using the Average Precision at Medium Intersection over Union (IoU) metric. The reported results showcase various algorithms' performance, with Dynamic Head (DyHead) emerging as the top scorer, excelling in accurately localizing and classifying objects. Its high precision and recall at medium IoU thresholds signify robustness. The paper suggests considering the mean Average Precision (mAP) metric for a comprehensive evaluation across IoU thresholds, if available. Despite this, DyHead stands out as the superior algorithm, particularly at medium IoU thresholds, making it suitable for precise object detection in smart city applications. The performance analysis using Average Precision at Medium IoU is reinforced by the Average Precision at Low IoU (APL), consistently depicting DyHead's superiority. These findings provide valuable insights for researchers and practitioners, guiding them toward employing DyHead for tasks prioritizing accurate object localization and classification in smart cities. Overall, the paper navigates through the complexities of object detection in urban environments, presenting DyHead as a leading solution with robust performance metrics.

Keywords: Object detection, Deep learning, Performance analysis, Accuracy, COCO dataset

Introduction

Object detection is a fundamental task in computer vision that involves locating and identifying multiple objects within an image or video [4]. It plays a crucial role in various applications, ranging from security surveillance and autonomous vehicles to augmented reality and robotics [5, 13, 27]. The ability to accurately detect and recognize objects in real-time is essential for enabling smart cities to efficiently manage and optimize their resources.

In the context of smart cities, object detection serves as a critical technology that lays the foundation for numerous intelligent systems [12, 13]. By leveraging computer vision and machine learning techniques [6, 20], smart cities can automate and enhance various aspects of urban life [21]. Object detection enables the monitoring of traffic flow, pedestrian movement, and vehicle identification, leading to more efficient traffic management and improved safety [15, 30]. Additionally, it aids in waste management by optimizing garbage collection routes through the identification of fill levels in trash bins. Moreover, object detection is instrumental in public safety and security [9, 16], as it allows for the detection of suspicious activities and potential threats in crowded areas or critical infrastructure [3]. These applications demonstrate the indispensable role of object detection in making urban environments more efficient, secure, and livable.

Over the years, significant progress has been made in object detection, driven mainly by advancements in deep learning techniques. Deep learning-based methods have garnered significant attention from researchers due to their ability to automatically learn relevant features from raw data [28, 31]. Compared to traditional approaches that heavily rely on handcrafted features, deep learning models offer superior performance and generalization capabilities [24]. Convolutional Neural Networks (CNNs) have revolutionized the field within various applications [2, 7], leading to remarkable improvements in detection accuracy and real-time processing capabilities. The development of Faster R-CNN [10], SSD (Single Shot Multibox Detector) [18], and YOLO (You Only Look Once) [25] are some of the milestone contributions that have paved the way for state-of-the-art object detection systems. Moreover, feature extraction methods like Region Proposal Networks (RPN) and Feature Pyramid Networks (FPN) have further improved the robustness and efficiency of object detection models.

While deep learning-based approaches have shown great promise in object detection, they still face several challenges, especially when applied to real-world smart city environments. One of the primary concerns is achieving high accuracy in complex and cluttered scenes with occlusions and varying lighting conditions. Additionally, meeting real-time requirements without sacrificing accuracy remains a significant challenge, considering the vast amounts of data processed in smart applications [1, 14]. As a result, further research and analysis are crucial to improve the performance of deep learning-based object detection algorithms and address these limitations.

This paper seeks to contribute to the field of deep learning-based object detection by conducting a thorough analysis of recent algorithms. It aims to identify the most accurate models for smart city applications, evaluate their real-time performance, and address the challenges they encounter. In this study, we explore each of the standard performance metrics involving Average Precision (AP), AP50, Average Precision Small (APS), Average Precision Medium (APM), and Average Precision Large (APL). Using these metrics helps us understand their significance in evaluating object detection algorithms on the widely used COCO benchmark [17]. Moreover, we analyze by illustrating the mAP results of object detection algorithms from 2016 onwards, based on Multi-scale, Single-scale, ResNet, FPN, DCN, and YOLO networks.

In this study, the research contributions are as follows,

1. Comprehensive analysis of deep learning algorithms: We conduct a detailed analysis of the most recent deep learning-based object detection algorithms, focusing on popular architectures such as Multi-scale, Single-scale, ResNet, FPN, DCN, and YOLO networks.
2. Identification of the most accurate models: Through extensive experimentation and evaluation, we identify the most accurate deep learning models for object detection in smart city scenarios.
3. Performance assessment under real-time constraints: We evaluate the shortlisted models' performance in real-time settings, considering the demanding requirements of smart city applications, to ascertain their feasibility and suitability for deployment.

Related works

This section reviews related works focusing on performance analysis studies on deep learning for object detection methods.

The paper in [22] provided a comprehensive examination of deep learning techniques for detecting small objects. The survey encompasses various state-of-the-art algorithms and evaluates their performance in this critical domain. The advantages of the paper lie in its thoroughness and systematic evaluation, which shed light on the strengths of different approaches in addressing the challenging task of detecting small objects accurately. By focusing on deep learning methods, the paper showcases the potential for significant advancements in small object detection, making it a valuable resource for researchers and practitioners in the field. However, one limitation could be the exclusivity to deep learning approaches, as it may overlook potential synergies with traditional methods or hybrid solutions. Nonetheless, the paper's contributions help advance the understanding and development of robust small object detection systems, facilitating applications in areas like surveillance, autonomous vehicles, and robotics.

In [19], a comprehensive exploration of various performance metrics was used to evaluate object detection algorithms. The survey encompasses a wide range of metrics, including Average Precision (AP), Intersection over Union (IoU), Precision-Recall curves, F1-score, and mAP (mean Average Precision). It provides a valuable resource for researchers and practitioners in the field, offering insights into the strengths and limitations of each metric and their applicability to different scenarios. By considering a diverse set of performance measures, the paper addresses the advantages of gaining a holistic understanding of an algorithm's capabilities and robustness in detecting objects accurately. Moreover, the survey helps researchers in choosing appropriate evaluation criteria based on their specific application requirements. However, one limitation of the paper could be the absence of a unified evaluation standard, as various metrics may be favored in different contexts, leading to challenges in comparing results across studies. Nevertheless, the paper's contributions serve to promote the development and benchmarking of more reliable and effective object detection algorithms, aiding advancements in computer vision and related applications.

In [29], a comprehensive overview of state-of-the-art object detection models is presented based on deep learning techniques. The survey covers a wide range of contemporary architectures and methodologies, such as Faster R-CNN, SSD, YOLO, and RetinaNet, providing valuable insights into their strengths and limitations. The advantages of the paper lie in its systematic analysis and comparison of various models, allowing readers to gain a deep understanding of their respective design choices, performance, and applicability in different scenarios. By consolidating information on these cutting-edge models, the paper serves as a valuable resource for researchers and practitioners in the field of computer vision, enabling them to make informed decisions when selecting or developing object detection systems. However, one limitation of the survey could be the dynamic nature of the field, with new models and techniques constantly emerging, potentially rendering some sections outdated over time. Nevertheless, the paper's contributions aid in driving advancements in object detection research, fostering the development of more accurate, efficient, and robust deep learning-based models for various real-world applications.

Methods

In the realm of computer vision and machine learning, object detection algorithms play a crucial role in identifying and localizing objects within images or video frames. Evaluating the performance of these algorithms is essential to gauge their accuracy and efficiency. In this study, we analyze the performance of object detection algorithms on a standard dataset and benchmark named COCO. One effective way to achieve this is by utilizing various performance metrics, such as Average Precision (AP), AP50, Average Precision Small (APS), Average Precision Medium (APM), and Average Precision Large (APL). In this study, we explore each of these metrics and understand their significance in evaluating object detection algorithms on the widely used COCO benchmark. Moreover, we analyze by illustrating the mAP results of object detection algorithms from 2016 onwards, based on Multi-scale, Single-scale, ResNet, FPN, DCN, and YOLO networks.

Average Precision (AP)

Average Precision (AP) is a fundamental metric used to measure the accuracy of an object detection algorithm. It calculates the average precision-recall curve over all classes in the dataset. Precision denotes the ratio of true positives to the sum of true positives and false positives, while recall represents the ratio of true positives to the sum of true positives and false negatives. AP provides an aggregate assessment of the algorithm's ability to correctly identify objects across different classes, making it a valuable tool for benchmarking performance.

AP50 and AP75

AP50 and AP75 are variants of the Average Precision metric, focusing on specific ranges of IoU (Intersection over Union) thresholds. IoU measures the spatial overlap between predicted and ground-truth bounding boxes. AP50 considers a threshold of 50%, whereas AP75 uses a threshold of 75%. These metrics are particularly useful when fine-tuning an object detection model for specific applications where high precision or recall is paramount.

APS, APM, and APL

The APS calculates the average precision only for small objects in the dataset. This metric provides insights into how well the algorithm handles small and challenging objects, which are often more difficult to detect accurately.

APM (Average Precision Medium): APM focuses on objects of medium size, aiming to assess the algorithm's performance on a different scale. Some objects might be relatively straightforward to detect when large but become more challenging when their size decreases.

APL (Average Precision Large): APL measures the average precision for large objects. This metric is essential for evaluating how well an algorithm can detect and localize larger objects, which often exhibit more complex and diverse shapes.

Performance metrics for evaluation

When evaluating object detection algorithms on the COCO benchmark, employing the aforementioned performance metrics is crucial to gaining comprehensive insights into their strengths and weaknesses. By analyzing AP, AP50, AP75, APS, APM, and APL, researchers and developers can fine-tune their models and optimize them for specific use cases.

In this evaluation, if an algorithm achieves high APS but lower APL, it might excel at detecting small objects but struggle with larger ones. On the other hand, an algorithm with high APM and APL but lower APS might perform better with larger and medium-sized objects but face difficulties in detecting smaller ones. Therefore, performance metrics such as Average Precision (AP), AP50, AP75, APS, APM, and APL are indispensable tools for evaluating the accuracy and efficiency of object detection algorithms. These metrics provide valuable insights into an algorithm's strengths and weaknesses across various object sizes and complexities. When using the COCO benchmark, researchers and developers can leverage these metrics to refine their models and optimize them for real-world applications.

Results and discussions

This section presents the results and details of the performance analysis of different object detection algorithms are discussed. In this study, we selected the most popular object detection based on reported results from corresponding studies as Faster R-CNN [10], Mask R-CNN [11], D-RFCN + SNIP [26], NAS-FPN [32], DetectorRS [23], and DyHead [8] algorithms.

COCO benchmark

The choice of the COCO (Common Objects in Context) dataset as the preferred dataset and benchmark for object detection is justified by its comprehensive and diverse nature. The COCO comprises a vast array of images spanning 80 object categories, capturing complex real-world scenarios with multiple objects in intricate spatial relationships. Its scale, diversity, and richness in contextual information make it an ideal testbed for evaluating the performance of object detection algorithms across various scenarios. Researchers widely adopt the COCO dataset due to its ability to challenge models with a wide spectrum of challenges, including occlusions, diverse object sizes, and crowded scenes, mirroring the complexities encountered in real-world applications. The popularity of COCO as a benchmark ensures that results obtained on this dataset are widely recognized and comparable, fostering a standardized evaluation framework across the computer vision community. Therefore, leveraging COCO as a benchmark dataset for object detection aligns with the need for a realistic and representative evaluation platform, facilitating meaningful comparisons and advancements in the field.

Object detection performance results

In our study, we present the performance of the popular object detection algorithms: Faster R-CNN, Mask R-CNN, D-RFCN + SNIP, DetectorRS, and DyHead, using a comprehensive set of performance metrics. These metrics included Average Precision (AP) across various Intersection over Union (IoU) thresholds, specifically AP at IoU = 0.50 (AP50), AP at IoU = 0.75 (AP75), AP for small objects (APs), AP for medium-sized objects (APm), and AP for large objects (APl). Our analysis revealed valuable insights into the algorithms' capabilities in detecting objects of different sizes and achieving accurate localization. This multi-faceted evaluation not only showcased their overall object detection prowess but also their effectiveness in handling diverse object scales, providing a holistic view of their performance in real-world scenarios. The results of these performance metrics offer a comprehensive basis for comparing and selecting the most suitable object detection algorithm for specific applications and use cases.

Figure 1 presents the performance result of the Faster R-CNN algorithm on different backbones, as measured by various performance metrics. In terms of the Average Precision (AP) metric, the highest value is achieved by the LIP-ResNet-101-MD with FPN backbone, which recorded an impressive AP of 43.9, indicating its superior ability to accurately detect objects in the dataset. On the other hand, the Inception-ResNet-v2 backbone achieved the lowest AP score of 34.7, suggesting comparatively less effective object detection performance. When considering AP50, which measures the precision at a stricter IoU threshold of 0.50, we observe a similar pattern.

As shown in Table 1, the LIP-ResNet-101-MD with FPN backbone leads the pack with an AP50 score of 65.7, highlighting its excellence in accurately localizing objects.

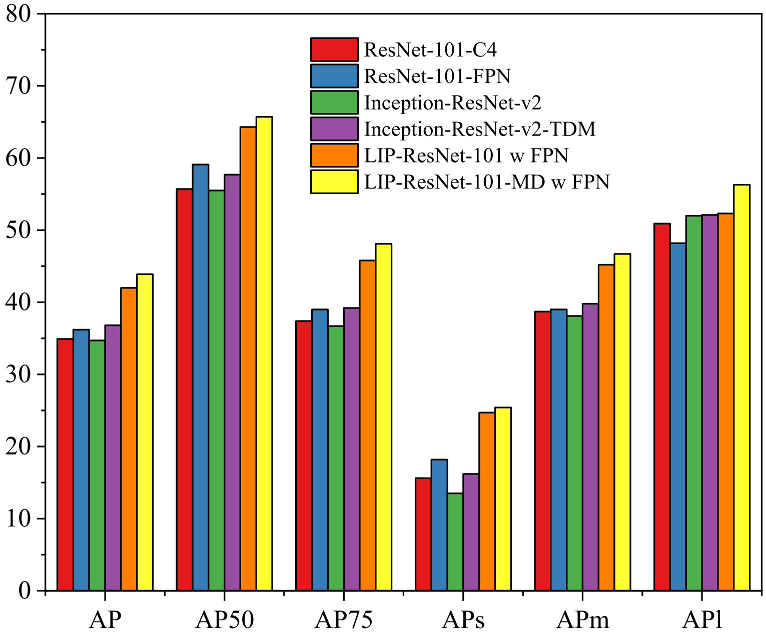

Fig. 1 Performance result of the Faster R-CNN algorithm on different backbones

In contrast, the Inception-ResNet-v2 backbone again obtains the lowest AP50 score of 55.5, indicating a comparatively lower precision in detecting objects at this IoU threshold. Moving on to AP75, which measures precision at a more stringent IoU threshold of 0.75, we see a similar trend. The LIP-ResNet-101-MD with FPN backbone outperforms all others with an AP75 score of 48.1, emphasizing its remarkable ability to precisely locate objects. Conversely, the Inception-ResNet-v2 backbone achieves the lowest AP75 score of 36.7, indicating less robust performance at this high IoU threshold.

Regarding the AP scores for small (APs), medium (APm), and large (APl) objects, the LIP-ResNet-101-MD with FPN backbone consistently attains the highest scores in all categories. This suggests that it excels in detecting objects of varying sizes. In contrast, the Inception-ResNet-v2 backbone consistently obtains the lowest scores in these categories. Therefore, based on the obtained results in Fig. 1, it is evident that the LIP-ResNet-101-MD with FPN backbone consistently outperforms the other backbones across all performance metrics, demonstrating its superiority in object detection capabilities. On the other hand, the Inception-ResNet-v2 backbone consistently ranks lower in performance, indicating that it may not be the best choice for tasks requiring precise and robust object detection. These findings highlight the importance of selecting an appropriate backbone when using the Faster R-CNN algorithm, as it can significantly impact the algorithm's performance in various scenarios and for objects of different sizes.

Figure 2 represents the performance results of the Mask R-CNN algorithm on different backbones, as evaluated by various performance metrics. Firstly, when examining the AP metric, it becomes apparent that the ResNeXt-101-FPN backbone attains the highest overall score of 39.8, indicating its exceptional ability to accurately detect objects in the dataset. Conversely, the Inception-ResNet-v2-TDM backbone records the lowest AP score of 36.8, suggesting comparatively less effective object detection performance.

For the AP50 metric, which measures precision at an IoU threshold of 0.50, we observe a similar trend. The ResNeXt-101-FPN backbone outperforms others with an AP50 score of 62.3, signifying its superior precision in detecting objects. On the other hand, the Inception-ResNet-v2-TDM backbone achieves the lowest AP50 score of 57.7, indicating a comparatively lower precision in object detection at this IoU threshold. Moving on to AP75, which evaluates precision at an IoU threshold of 0.75, the ResNeXt-101-FPN backbone continues to lead with an AP75 score of 43.4,

Table 1 Result of algorithms on different backbones

Method	AP	AP50	AP75	APs	APm	APl
ResNet-101-C4	34.8	55.5	36.7	14.9	39.6	52.1
ResNet-101-FPN	36.2	59.1	39	16.6	39.8	49.4
Inception-ResNet-v2	37.4	55.5	35.8	13.6	38.7	53.6
Inception-ResNet-v2-TDM	38.8	56.2	39.3	16.8	40.1	53.8
LIP-ResNet-101 w FPN	32.5	64.9	45.4	23.2	46.7	54
LIP-ResNet-101-MD w FPN	43.8	65.7	47.2	25.8	47.8	56.7

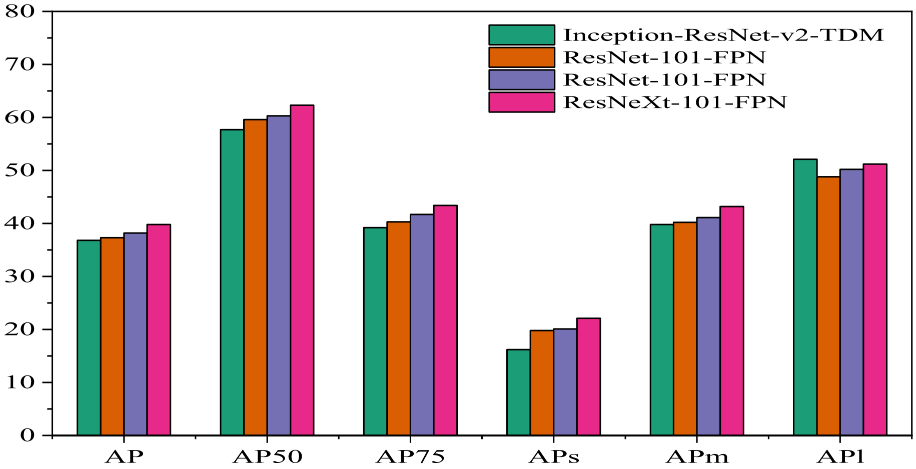

Fig. 2 Performance result of the Mask R-CNN algorithm on different backbones

emphasizing its excellent performance in precisely locating objects. Conversely, the Inception-ResNet-v2-TDM backbone records the lowest AP75 score of 39.2, indicating a relative struggle to achieve high precision at this stringent IoU threshold.

Based on the reported results, the ResNeXt-101-FPN backbone consistently outperforms others in all categories, reaffirming its strength in detecting objects of varying sizes. In contrast, the Inception-ResNet-v2-TDM backbone consistently obtains the lowest scores in these categories, suggesting limitations in handling different object scales. Therefore, based on reported, it is evident that the ResNeXt-101-FPN backbone consistently outperforms the other backbones across all performance metrics, demonstrating its superiority in object detection capabilities when paired with the Mask R-CNN algorithm. Conversely, the Inception-ResNet-v2-TDM backbone consistently ranks lower in performance, indicating that it may not be the best choice for tasks requiring precise and robust object detection. These findings emphasize the critical importance of selecting the appropriate backbone for the Mask R-CNN algorithm, as it can significantly impact performance across various scenarios and object characteristics.

Figure 3 presents the results of the D-RFCN + SNIP algorithm's performance across different backbone configurations, as measured by various performance metrics. In terms of the AP metric, the D-RFCN + SNIP algorithm with the DPN-98 backbone (with flip and multi-scale) emerges as the top performer, achieving an impressive AP score of 45.7. This result indicates its exceptional ability to accurately detect objects in the dataset. On the other hand, the D-RFCN + SNIP algorithm with the ResNet-101 backbone (multi-scale) achieves a slightly lower AP score of 43.4, suggesting a slightly less effective object detection performance in comparison. Examining the AP50 metric, which measures precision at an IoU threshold of 0.50, we again see the D-RFCN + SNIP algorithm with the DPN-98 backbone outperforming its counterpart with a score of 67.3, indicating superior precision in object detection. In contrast, the D-RFCN + SNIP algorithm with the ResNet-101 backbone obtains an AP50 score of 65.5, demonstrating a slightly lower precision at this IoU threshold.

Similarly, the D-RFCN + SNIP algorithm with the DPN-98 backbone leads in the AP75 metric with a score of 51.1, highlighting its remarkable precision at a more

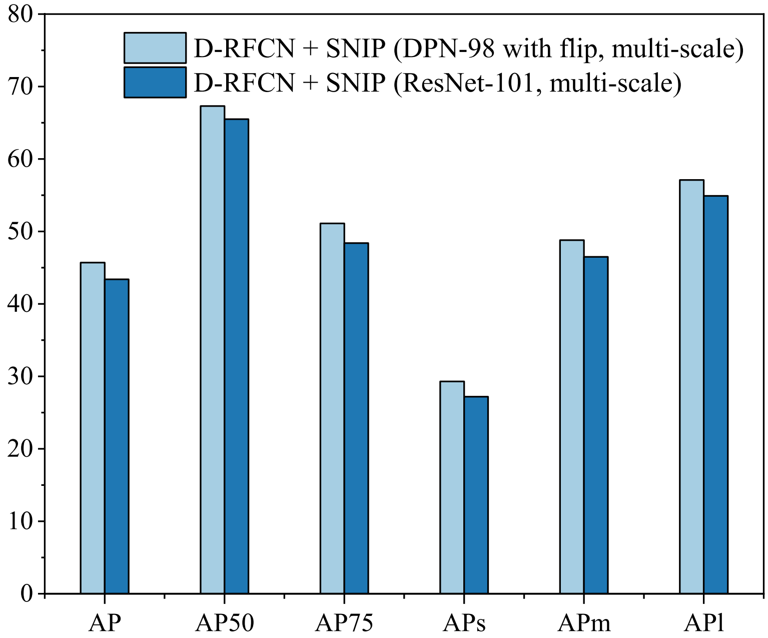

Fig. 3 Performance result of the D-RFCN + SNIP algorithm on different backbones

stringent IoU threshold of 0.75. Conversely, the D-RFCN + SNIP algorithm with the ResNet-101 backbone records a lower AP75 score of 48.4, indicating a slightly reduced ability to achieve high precision at this stringent IoU threshold. Therefore, the results demonstrate that the D-RFCN + SNIP algorithm with the DPN-98 backbone consistently outperforms its counterpart with the ResNet-101 backbone across all performance metrics, highlighting its superiority in object detection capabilities. This underscores the significance of selecting the appropriate backbone when implementing the D-RFCN + SNIP algorithm, as it can significantly influence performance in various scenarios and object detection characteristics. Researchers and practitioners should consider the specific demands of their object detection tasks when choosing a backbone, with the DPN-98 backbone showcasing exceptional performance for tasks requiring accurate and high-precision object detection.

Figure 4 presents the results of the DetectorRS algorithm's performance with various backbone configurations, as measured by different performance metrics. Similar to the previous interpretation, in terms of the AP metric, the DetectorRS algorithm with the ResNeXt-101–64 × 4d backbone stands out as the top performer, achieving the highest AP score of 48.5. This indicates its exceptional ability to accurately detect objects in the dataset. On the other hand, the DetectorRS algorithm with the ResNet-50 backbone records the lowest AP score of 44.4, suggesting comparatively less effective object detection performance. When examining the AP50 metric, which measures precision at an IoU threshold of 0.50, we again observe that the DetectorRS algorithm with the ResNeXt-101–64 × 4d backbone excels with an AP50 score of 72, indicating superior precision in object detection. In contrast, the DetectorRS algorithm with the ResNet-50 backbone obtains a lower AP50 score of 67.7, signifying a comparatively lower precision at this IoU threshold.

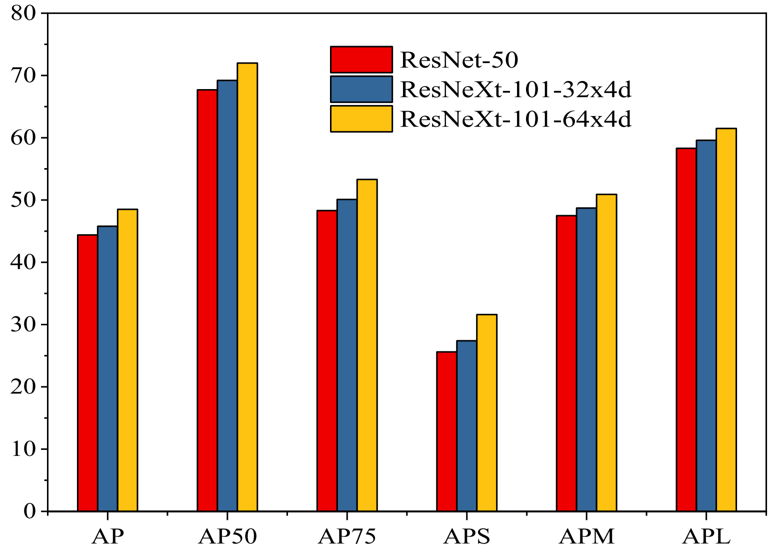

Fig. 4 Performance result of the DetectorRS algorithm on different backbones

Similarly, the DetectorRS algorithm with the ResNeXt-101–64 × 4d backbone leads in the AP75 metric with a score of 53.3, emphasizing its remarkable precision at a more stringent IoU threshold of 0.75. Conversely, the DetectorRS algorithm with the ResNet-50 backbone records the lowest AP75 score of 48.3, indicating a slightly reduced ability to achieve high precision at this stringent IoU threshold.

As a result, the results indicate that the DetectorRS algorithm with the ResNeXt-101–64 × 4d backbone consistently outperforms its counterparts with other backbones across all performance metrics, highlighting its superiority in object detection capabilities. This underscores the significance of selecting the appropriate backbone when implementing the DetectorRS algorithm, as it significantly influences performance in various scenarios and object detection characteristics. Researchers and practitioners should consider the specific requirements of their object detection tasks when choosing a backbone, with the ResNeXt-101–64 × 4d backbone demonstrating exceptional performance for tasks demanding accurate and high-precision object detection.

Figure 5 demonstrates the results of the DyHead algorithm's performance across different backbone configurations. The DyHead algorithm with the ResNeXt-64 × 4d-101-DCN backbone emerges as the top performer, achieving the highest AP score of 54.0. This indicates its remarkable ability to accurately detect objects within the dataset. Conversely, the DyHead algorithm with the ResNet-50 backbone records the lowest AP score of 43.0, suggesting a comparatively less effective object detection performance. When considering the AP50 metric, which measures precision at an IoU threshold of 0.50, we observe a similar trend. The DyHead algorithm with the ResNeXt-64 × 4d-101-DCN backbone excels with an AP50 score of 72.1, signifying superior precision in object detection. On the other hand, the DyHead algorithm with the ResNet-50 backbone again records the lowest AP50 score of 60.7, indicating a relatively lower precision at this IoU threshold.

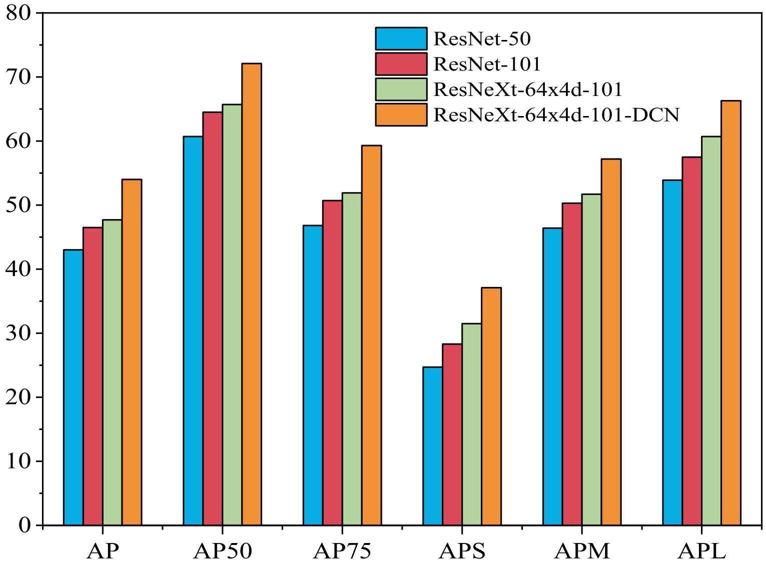

Fig. 5 Performance result of the DyHead algorithm on different backbones

Furthermore, the DyHead algorithm with the ResNeXt-64 × 4d-101-DCN backbone leads in the AP75 metric with a score of 59.3, emphasizing its outstanding precision at a more stringent IoU threshold of 0.75. In contrast, the DyHead algorithm with the ResNet-50 backbone records the lowest AP75 score of 46.8, indicating a somewhat reduced ability to achieve high precision at this stringent IoU threshold.

As a result, the results represent that the DyHead algorithm with the ResNeXt-64 × 4d-101-DCN backbone consistently outperforms other backbone configurations across all performance metrics, demonstrating its superiority in object detection capabilities. This highlights the importance of selecting the appropriate backbone when implementing the DyHead algorithm, as it plays a pivotal role in influencing performance across different scenarios and object detection characteristics. Researchers and practitioners should consider the specific demands of their object detection tasks when making backbone choices, with the ResNeXt-64 × 4d-101-DCN backbone showcasing exceptional performance for tasks demanding precise and high-precision object detection.

Performance analysis based on the mAP metric

As mentioned earlier, one commonly used metric for performance evaluation of object detection algorithms is mean Average Precision (mAP). The mAP measures the accuracy of the model in localizing and classifying objects within an image. It takes into account precision and recall values at various intersection over union (IoU) thresholds, which determine how well the predicted bounding boxes overlap with ground-truth bounding boxes.

In short description, Faster R-CNN is a well-known two-stage object detection framework that uses a Region Proposal Network (RPN) to generate candidate object regions, followed by classification and bounding box regression. Mask R-CNN extends Faster R-CNN by adding a mask prediction branch, enabling instance segmentation along with object detection. D-RFCN + SNIP is a method that combines the advantages of D-RFCN, which dynamically adjusts the receptive field size, and SNIP, a network pruning technique, to achieve better efficiency and accuracy. NAS-FPN stands for Neural Architecture Search for Feature Pyramid Network, a method that automatically searches for the optimal architecture of the feature pyramid used in object detection models. DetectorRS is an algorithm that focuses on enhancing object detection in remote sensing images. Lastly, DyHead refers to Dynamic Head, which dynamically predicts the number of object instances in an image without fixed anchor priors. Figure 1 shows the performance comparison using the mAP metric for the object detection algorithms.

As illustrated in Fig. 6, the reported results in the graph show the performance evaluation of various object detection algorithms in terms of box mAP. The mAP values for each algorithm are as follows: Faster R-CNN achieved 34.90, Mask R-CNN obtained 39.80, D-RFCN + SNIP scored 45.70, NAS-FPN achieved 50.70, DetectorRS attained 55.70, and DyHead obtained the highest score of 60.60.

Based on the obtained mAP values, it is evident that DyHead is the superior algorithm among the compared ones. With a box mAP of 60.60, DyHead outperforms all other algorithms, achieving the highest accuracy in localizing and classifying objects within images. NAS-FPN also performs well, obtaining the second-highest mAP score of 50.70, followed by DetectorRS at 55.70. D-RFCN + SNIP and Mask R-CNN have mAP scores of 45.70 and 39.80, respectively, indicating moderate performance. Faster R-CNN trails behind the others, achieving the lowest mAP score of 34.90.

Therefore, based on the reported results and the mAP metric, DyHead is the most effective algorithm for object detection, offering the highest accuracy and precision among the compared models. Researchers and practitioners looking for state-of-the-art performance in object detection tasks should consider using DyHead as their preferred choice.

Performance analysis based on AP50

AP50 is an intersection over union (IoU) threshold of 50%, which measures the accuracy of the algorithms in localizing and classifying objects with a moderate overlap between predicted bounding boxes and ground-truth bounding boxes. The algorithms compared in the chart are Mask R-CNN, D-RFCN + SNIP, Cascade Mask R-CNN, DetectorRS,

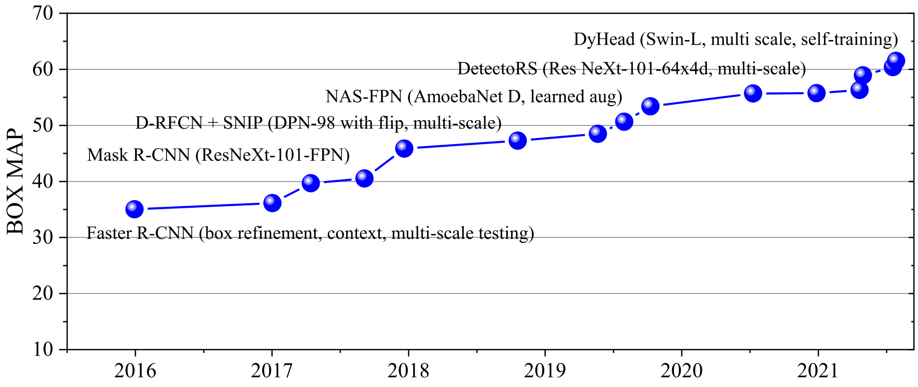

Fig. 6 Performance comparison using the AP50 metric

and DyHead. By analyzing the AP50 scores for these algorithms, we can gain insights into their effectiveness in detecting objects accurately when there is a reasonable overlap between the predicted and actual bounding boxes. Figure 2 shows the performance comparison using the AP50 metric for the object detection algorithms.

As shown in Fig. 7, the reported results in the graph show the performance evaluation of different object detection algorithms using the AP50 metric. The AP50 values for each algorithm are as follows: Mask R-CNN achieved 62.30, D-RFCN + SNIP obtained 67.30, Cascade Mask R-CNN scored 71.90, DetectorRS achieved 74.20, and DyHead obtained the highest score of 78.50.

Based on the reported results, it is evident that DyHead is the superior algorithm among the compared ones. With an AP50 score of 78.50, DyHead outperforms all other algorithms, achieving the highest accuracy in localizing and classifying objects when there is a moderate overlap between predicted bounding boxes and ground-truth bounding boxes. DetectorRS also performs well, obtaining the second-highest AP50 score of 74.20, followed by Cascade Mask R-CNN at 71.90. D-RFCN + SNIP and Mask R-CNN have AP50 scores of 67.30 and 62.30, respectively, indicating moderate performance.

Performance analysis using APM and APL

The performance evaluation metrics APM and AP are commonly used in the context of object detection to assess the accuracy and robustness of different algorithms. These metrics measure the performance of the algorithms at different intersection over union (IoU) thresholds, which determine the extent of overlap between predicted bounding boxes and ground-truth bounding boxes. APM focuses on medium IoU thresholds, typically ranging from 0.5 to 0.75. This range represents a moderate level of overlap, where the predicted bounding boxes are considered accurate if they have a reasonable match with the ground-truth bounding boxes. APM evaluates how well the algorithms can accurately localize and classify objects with moderate spatial agreement. It is especially useful in scenarios where precise localization is essential but allowing some flexibility in the bounding box predictions. On the other hand, APL considers large IoU thresholds, usually greater than 0.75. This higher threshold requires a much closer alignment between the predicted and ground-truth bounding boxes. APL evaluates the performance of the algorithms in detecting and classifying objects with a high degree of

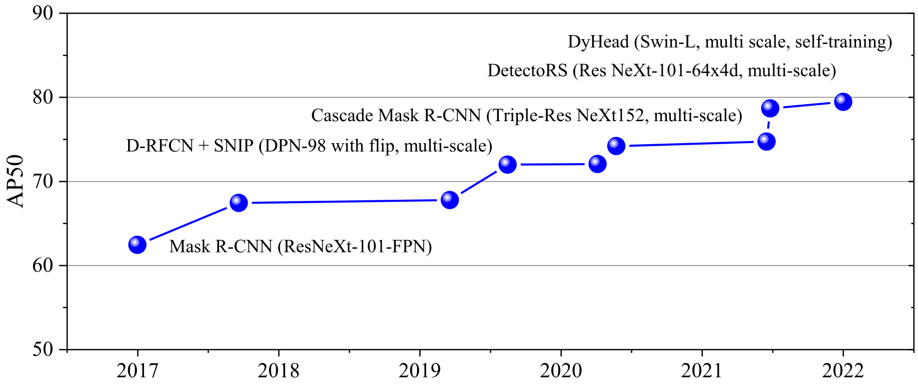

Fig. 7 Performance evaluation using the AP50 metric

precision. It is particularly relevant in scenarios where precise localization is crucial, and the algorithm needs to be robust to variations in object positions and sizes.

As shown in Fig. 8, the reported results in the graph present the performance evaluation of various object detection algorithms using the APM (Average Precision at Medium IoU) metric. The APM values for each algorithm are as follows: Mask R-CNN achieved 43.20, D-RFCN + SNIP obtained 48.80, PANet scored 51.70, NAS-FPN achieved 55.50, DetectorRS attained 58.40, and DyHead obtained the highest score of 64.00.

Based on the reported results and the APM metric, DyHead is the superior algorithm among the compared ones. With an APM score of 64.00, DyHead outperforms all other algorithms in accurately localizing and classifying objects with a moderate level of overlap between predicted and ground-truth bounding boxes. It achieves the highest precision and recall at medium IoU thresholds, indicating robustness and accuracy in object detection tasks. While DyHead is the superior algorithm based on APM, it is essential to consider its performance in relation to the mAP metric, which provides a more comprehensive evaluation across different IoU thresholds. mAP is generally a more widely used and informative metric for object detection models. If the reported mAP values are available for these algorithms, it would be more appropriate to analyze and compare their performance using that metric.

Nonetheless, based on the reported APM results, DyHead stands out as the top-performing algorithm for object detection at medium IoU thresholds. Its higher APM score compared to other algorithms, such as DetectorRS and NAS-FPN, demonstrates its superior ability to accurately detect and classify objects under moderate spatial agreement. Researchers and practitioners may consider using DyHead for tasks that prioritize precise localization and classification of objects in object detection applications. As reported in the performance analysis using APM, the APL presented a similar result that indicates the DyHead presents better results compared to others as shown in Fig. 9.

Discussions and considerations

It is imperative to acknowledge potential biases within the COCO dataset, as its composition may not perfectly mirror the diverse range of demographic and environmental conditions found in real-world smart city applications. Biases, whether in terms of object representation or contextual scenarios, could impact the generalizability of the

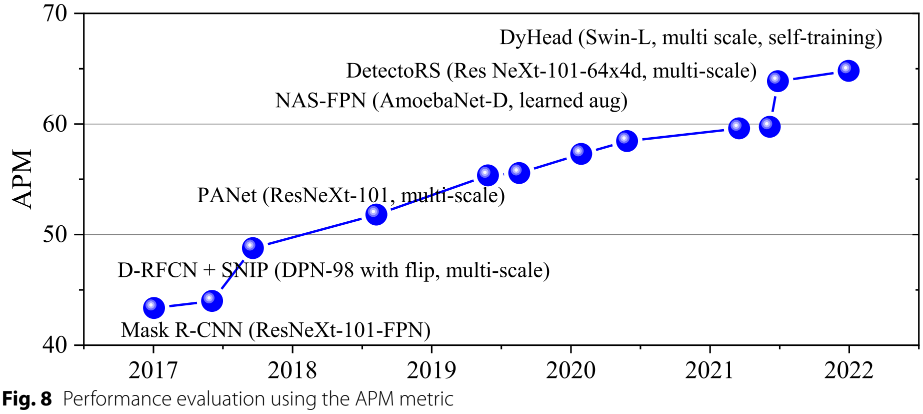

Fig. 8 Performance evaluation using the APM metric

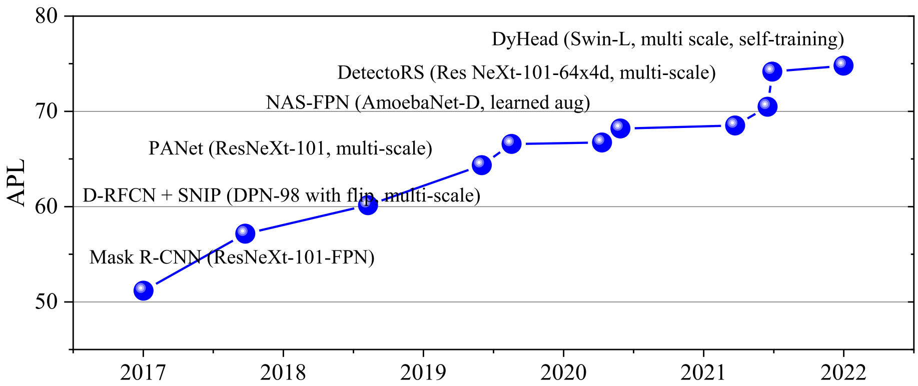

Fig. 9 Performance evaluation using the APL metric

study's findings. Researchers should be mindful of these limitations to ensure that the insights gained from object detection algorithms on COCO are appropriately interpreted and applied to the complexities of varied urban environments.

The computational demands of advanced algorithms, exemplified by DyHead, must be highlighted, considering their potential limitations for real-time deployment on resource-constrained devices commonly found in smart city infrastructures. Practicality is a crucial aspect of algorithm selection for urban applications, and understanding the computational resource requirements helps manage expectations regarding the feasibility of implementing such algorithms in real-world settings.

While widely accepted metrics like Average Precision (AP) and mean Average Precision (mAP) provide valuable benchmarks for assessing object detection algorithms, it is essential to recognize that these metrics might not capture all aspects of performance in the intricacies of complex urban environments or specific scenarios. Supplementary metrics or context-specific evaluations may be necessary to provide a more nuanced understanding of algorithmic effectiveness in diverse smart city contexts.

Addressing the limitations of chosen algorithms or architectures with respect to certain types of objects or environmental conditions is crucial. An analysis that considers the scope of each algorithm's applicability provides a more accurate portrayal of their strengths and weaknesses. Understanding which scenarios or object types a particular algorithm excels or struggles with enhances the practicality of its implementation in smart city contexts.

Bridging the gap between research findings and the challenges of real-world implementation in smart cities is paramount. Scalability, real-time responsiveness, and integration complexities should be discussed to elucidate the practical implications of deploying object detection algorithms in dynamic urban environments. Acknowledging these challenges ensures a more holistic understanding of the algorithm's utility and feasibility for actual smart city applications.

As a result, while the study offers a comprehensive exploration of object detection algorithms in the context of smart cities, it is crucial to acknowledge several potential drawbacks in the approach taken. Firstly, the reliance on the COCO dataset, while widely accepted, introduces biases that may limit the generalizability of the findings to diverse urban environments. The dataset's representation might not fully encapsulate the complexities of specific smart city scenarios, raising questions about the algorithms' adaptability to a broader range of real-world situations. Additionally, the computational demands of advanced algorithms, exemplified by DyHead, may pose practical challenges for real-time deployment in resource-constrained smart city infrastructures. This consideration is vital for ensuring that the proposed solutions are not only effective but also feasible for implementation in practical settings. Moreover, while established metrics like AP and mAP provide valuable benchmarks, they might not comprehensively capture the intricate challenges posed by complex urban environments, emphasizing the need for additional context-specific evaluations to holistically assess algorithmic performance. The study's scope may also have limitations in analyzing the applicability of the chosen algorithms to specific types of objects or environmental conditions, potentially overlooking critical factors that influence their effectiveness. Finally, there is a need to address the gap between research findings and real-world implementation challenges in smart cities, considering issues like scalability and integration complexities. Recognizing these potential drawbacks is essential for a nuanced understanding of the study's limitations and for guiding future research toward more robust and applicable solutions for enhancing object detection in smart city environments.

Conclusions

This conclusion paper provides significant contributions to the field of deep learning-based object detection by conducting a thorough analysis of recent algorithms. It aims to identify the most accurate models suitable for smart city applications, evaluating their real-time performance and addressing challenges. The study explores standard performance metrics such as Average Precision (AP), AP50, APS, APM, and APL to assess object detection algorithms using the COCO benchmark. Additionally, it analyzes the mAP results of object detection algorithms from 2016 onwards, focusing on popular architectures like Multi-scale, Single-scale, ResNet, FPN, DCN, and YOLO networks. The research offers comprehensive insights into the strengths and limitations of these algorithms, contributing to the advancement of intelligent urban infrastructures and safety through improved object detection systems. Future work suggestions include optimizing models for even faster real-time inference and deploying them efficiently on resource-constrained devices. Additionally, there is potential in developing adaptive models to address dynamic changes in smart city environments, ensuring the continued effectiveness of object detection systems as cities evolve. This research contributes to advancing intelligent urban infrastructures and safety through improved object detection capabilities.

Acknowledgements
This work was supported by "Sponsored by Natural Science Foundation of Xinjiang Uygur Autonomous Region(2022D01C461)," "Research on UAV moving target detection and tracking system based on computer vision (ZZ202105)," and "Research on image recognition technology of tomato diseases in Xinjiang Based on deep learning(2022D01C349)."

Authors' contributions
All authors contributed to the study's conception and design. Data collection, simulation, and analysis were performed by Jiya Tian, Qiangshan Jin, Yizong Wang, Jie Yang, Shuping Zhang, and Dengxun Sun. The first draft of the manuscript was written by Jiya Tian and all authors commented on previous versions of the manuscript.
All authors have read and approved the manuscript.

Funding
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials

Data can be shared upon request.

Declarations

Competing interests

The authors declare no competing interests.

Received: 15 October 2023 Accepted: 5 March 2024
Published online: 21 March 2024

References

1. Aghamohammadi A, Ang MC, Sundararajan EA et al (2018) A parallel spatiotemporal saliency and discriminative online learning method for visual target tracking in aerial videos. PLoS ONE 13:e0192246
2. Aghamohammadi A, Beheshti Shirazi SA, Banihashem SY, et al (2023) A deep learning model for ergonomics risk assessment and sports and health monitoring in self-occluded images. Signal, Image and Video Processing (SIViP). Springer, 18:1161–1173. https://doi.org/10.1007/s11760-023-02830-6
3. Alshammari A, Rawat DB (2019) Intelligent multi-camera video surveillance system for smart city applications, Computing and Communication Workshop and Conference (CCWC). IEEE, p 0317–0323
4. Amit Y, Felzenszwalb P (2014) Object Detection. In: Ikeuchi K. (eds) Computer Vision. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-31439-6_660
5. Ang M, Sundararajan E, Ng K et al (2014) Investigation of threading building blocks framework on real time visual object tracking algorithm. Appl Mech Mater 666:240–244
6. Ang MC, Aghamohammadi A, Ng KW et al (2014) Multi-core frameworks investigation on a real-time object tracking application. Journal of Theoretical and Applied Information Technology 70(1):163–171
7. Arabi S, Haghighat A, Sharma A (2020) A deep-learning-based computer vision solution for construction vehicle detection. Computer-Aided Civil and Infrastructure Engineering 35:753–767
8. Dai X, Chen Y, Xiao B, et al (2021) Dynamic head: unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 7373–7382
9. Elsaeidy A, Munasinghe KS, Sharma D, Jamalipour A (2019) Intrusion detection in smart cities using Restricted Boltzmann Machines. J Netw Comput Appl 135:76–83
10. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision. p 1440–1448
11. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. p 2961–2969
12. Hu L, Ni Q (2017) IoT-driven automated object detection algorithm for urban surveillance systems in smart cities. IEEE Internet Things J 5:747–754
13. Ingle PY, Kim Y-G (2022) Real-time abnormal object detection for video surveillance in smart cities. Sensors 22:3862
14. Jiang Z, Zhao L, Li S, Jia Y (2020) Real-time object detection method based on improved YOLOv4-tiny. arXiv preprint arXiv: 2011.04244
15. Khan S, Teng Y, Cui J (2021) Pedestrian traffic lights classification using transfer learning in smart city application. In: 2021 13th International conference on communication software and networks (ICCSN). IEEE, p 352–356
16. Laufs J, Borrion H, Bradford B (2020) Security and the smart city: a systematic review. Sustain Cities Soc 55:102023
17. Lin T-Y, Maire M, Belongie S et al (2014) Microsoft COCO: Common Objects in Context. In: Computer Vision–ECCV 2014, 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. Springer, p 740–755
18. Liu W, Anguelov D, Erhan D et al (2016) SSD: single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, p 21–37
19. Liu Y, Sun P, Wergeles N, Shang Y (2021) A survey and performance evaluation of deep learning methods for small object detection. Expert Syst Appl 172:114602
20. Mogharrebi M, Ang MC, Prabuwono AS et al (2013) Retrieval system for patent images. Procedia Technol 11:912–918
21. Muthanna MSA, Lyachek YT, Musaeed AMO et al (2020) Smart system of a real-time pedestrian detection for smart city. In: 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, p 45–50
22. Padilla R, Netto SL, Da Silva EA (2020) A survey on performance metrics for object-detection algorithms. In: 2020 international conference on systems, signals and image processing (IWSSIP). IEEE, p 237–242
23. Qiao S, Chen L-C, Yuille A (2021) Detectors: detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p 10213–10224
24. Ranjbarzadeh R, Ghoushchi SJ, Bendechache M et al (2021) Research article lung infection segmentation for COVID-19 pneumonia based on a cascade convolutional network from CT images.
25. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 779–788
26. Singh B, Davis LS (2018) An analysis of scale invariance in object detection snip. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p 3578–3587
27. Wang L, Sng D (2015) Deep learning algorithms with applications to video analytics for a smart city: a survey. arXiv preprint arXiv:1512.03131

28. Xiao Y, Tian Z, Yu J et al (2020) A review of object detection based on deep learning. Multimedia Tools and Applications 79:23729–23791
29. Zaidi SSA, Ansari MS, Aslam A et al (2022) A survey of modern deep learning based object detection models. Digital Signal Processing 126:103514
30. Zhang H, Du Y, Ning S et al (2017) Pedestrian detection method based on Faster R-CNN. In: 2017 13th International Conference on Computational Intelligence and Security (CIS). IEEE, p 427–430
31. Zhao Z-Q, Zheng P, Xu S-T, Wu X (2019) Object detection with deep learning: a review. IEEE transactions on neural networks and learning systems 30:3212–3232
32. Zoph B, Cubuk ED, Ghiasi G et al. (2020) Learning data augmentation strategies for object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16. Springer, p 566–583

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.