

Head Counting in Crowded Scenes Using YOLOv10: A Deep Learning Approach

Raghavendra V Vadavadagi^a, Sukul E N^b, Ankush Marlinganavvar^c, Anurag Hurkadli^d,
Kunal Bhoomraddi^e and Uday Kulkarni

School of Computer Science and Engineering, KLE Technological University, Hubballi, India

Keywords: YOLOv10, Head Counting, Object Detection, Deep Learning, Crowd Counting.


Abstract: Crowd counting plays a critical role in various applications, including public safety, event management, and resource planning, by accurately estimating the number of individuals in crowded environments. This study explores the use of the advanced YOLOv10 object detection framework for counting people in such settings. By leveraging image augmentation techniques, the dataset was enhanced to improve the model's robustness and ability to handle challenges like occlusion, overlapping objects, and varying lighting conditions. The YOLOv10 model demonstrated strong performance, achieving 49% validation accuracy at an IoU of 0.5 and 39% accuracy across IoU thresholds ranging from 0.5 to 0.9. These results underscore the model's effectiveness in real-world crowd detection, even under complex circumstances. The model's real-time detection capability makes it highly suitable for surveillance systems and other applications with limited computational resources. By integrating YOLOv10 into such systems, this work offers a scalable, efficient solution for accurate crowd counting, supporting safer and more efficient management of crowded scenarios. The model's potential for further improvements, such as hyperparameter tuning, extended training, and data augmentation, promises even greater performance and scalability in future deployments.


1 INTRODUCTION


Object detection is a basic task in the field of computer vision, with very broad applications in autonomous driving, video surveillance, and other domains. Among various deep learning algorithms developed for object detection, the YOLO(Shi et al.(2023)Shi, Wang, and Guo) family of models probably emerged as one of the most effective approaches that balance accuracy and speed. YOLO models remain single-stage detectors, in the sense that they predict both the bounding box coordinates and class probabilities of objects in one pass through the network itself, making them extremely fast compared to two-stage detectors such as Faster Recurrent convolution neural network (RCNN)(Xie et al.(2021)Xie, Cheng, Wang, Yao, and Han) . The architecture of YOLO is designed to process an image in a whole


manner to enable real-time object detection with no great loss for accuracy.


YOLOv1(Terven et al.(2023)Terven, Córdoba-Esparza, and Romero-González) was an innovative architecture for object detection. This model also suffered from some weaknesses: it did not work well for small objects and could not handle complex backgrounds. In the further versions of YOLO(Diwan et al.(2023)Diwan, Anirudh, and Tembhurne), namely YOLOv2(Sang et al.(2018)Sang, Wu, Guo, Hu, Xiang, Zhang, and Cai) and YOLOv3(Fu et al.(2018)Fu, Wu, and Zhao)(Balamurugan et al.(2021)Balamurugan, Santhanam, Billa, Aggarwal, and Alluri), several techniques were introduced, such as batch normalization, anchor boxes, and multi-scale prediction, which improved the accuracy. Meanwhile, YOLOv4 and YOLOv5(Olorunshola et al.(2023)Olorunshola, Irhebhude, and Ewwiek-paefe)(Bochkovskiy et al.(2020)Bochkovskiy, Wang, and Liao) focused more on enhancing the detection performance of previous versions for a faster speed of models, especially on edge devices. Afterwards, in an effort to further optimize performance and accuracy with more balanced computational cost, even more

^a  <https://orcid.org/0009-0009-4469-2874>

^b  <https://orcid.org/0009-0008-8841-7009>

^c  <https://orcid.org/0009-0008-8841-7009>

^d  <https://orcid.org/0009-0008-8841-7009>

^e  <https://orcid.org/0009-0008-7452-142X>

comprehensive architecture-based optimizations and feature enhancements were made in YOLOv6 and YOLOv7(Ajitha Gladis et al.)(Ajitha Gladis, Srinivasan, Sugashini, and Ananda Raj)(Sajitha et al.(2023)Sajitha, Andrushia, and Suni)

For instance, YOLOv8(Karthika et al.(2024)Karthika, Dharssinee, Reshma, Venkatesan, and Sujarani) achieved major improvements in performance thanks to enhanced model structure, loss functions, and inference strategies. Being efficient did not guarantee that YOLOv8(Karthika et al.(2024)Karthika, Dharssinee, Reshma, Venkatesan, and Sujarani) would have been able to detect objects from a highly cluttered scene—for example, in an extremely crowded environment.

Amongst the series, the latest, YOLOv10(Li et al.(2022)Li, Li, Jiang, Weng, Geng, Li, Ke, Li, Cheng, Nie, et al.), boasts a good number of significant enhancements both in performance and effect for crowd counting and activities alike that require small or highly tagged target detection. YOLOv10 has incorporated improved feature extraction layers with mechanisms of attention, allowing significant improvements in the focusing on relevant features in highly complex scenes. It has been optimized in terms of small object detection and occlusion management, making it much better than others insofar as crowd-like scenarios are concerned. The architecture design of YOLOv10(Sudharson et al.(2023)Sudharson, Srinithi, Akshara, Abhirami, Sriharshitha, and Priyanka) is in such a way that it maintains high accuracy while operating in real time, hence making it ideal for tasks requiring both speed and precision.

Crowd counting (Ruchika et al.(2022)Ruchika, Purwar, and Verma)is an important real-world application of object detection. Estimation of the number of people in crowded situations serves a variety of purposes: for event management, public safety, resource allocation, and so on. However, the traditional methods of counting usually involve either manual counting or simple computer vision approaches, neither of which scale well and are only moderately accurate in highly crowded situations where occlusion and other overlaps of objects may be frequent. In particular, automated head detection becomes vital in the use of surveillance systems with regard to crowd analyses where interest may lie in assessing or estimating crowd behavior, density, and motion patterns.

This paper mainly deals with head detection and crowd counting using YOLOv10(Li et al.(2022)Li, Li, Jiang, Weng, Geng, Li, Ke, Li, Cheng, Nie, et al.). The robustness of YOLOv10(Li et al.(2022)Li, Li, Jiang, Weng, Geng, Li, Ke, Li, Cheng, Nie, et al.)

against crowded scenes where the heads are overlapping or partial occlusion can estimate the correct crowd. With growing urbanization and the need to monitor large gatherings, effective crowd counting systems become critical concerning public safety, event management, and resource planning. In relation to this, our study will present how YOLOv10 can take up these challenges by providing a scalable and robust head detection in dense environments.”

This study showcases YOLOv10’s effectiveness in tackling crowd counting challenges, achieving a validation accuracy of 49% at IoU 0.5 and 39% across IoU thresholds from 0.5 to 0.9. These results highlight the model’s robustness and real-time detection capabilities, making it suitable for surveillance and efficient crowd management in real-world scenarios.

The paper is organized as follows: Section II provides a background study on object detection and previous advancements in the YOLO family, particularly focusing on YOLOv10. Section III describes the proposed methodology for head counting in crowded environments, detailing the architecture, loss function, and design considerations. Section IV presents the experimental results and analysis of the model’s performance. Finally, Section V concludes the paper and discusses potential future work to enhance the model’s accuracy and scalability.

2 BACKGROUND STUDY

Head counting using computer vision has become an essential area of research due to its applications in crowd management, public safety, and behavioral analysis. The YOLO (You Only Look Once)(Lin and Sun(2018)) algorithm has proven to be highly effective for such tasks, especially because of its ability to detect objects quickly and accurately in real-time. Unlike older methods that rely on multiple steps like region proposal and classification, YOLO simplifies the process by analyzing the entire image in a single step. This approach makes it suitable for dynamic and crowded environments where speed and precision are crucial.

YOLO(Jiang et al.(2022)Jiang, Ergu, Liu, Cai, and Ma) works by dividing an image into a grid of cells. Each cell predicts bounding boxes, confidence scores, and class probabilities for objects in its area. The confidence scores indicate how likely it is that an object is present in a bounding box and how accurate the predicted box is. One of the key advancements in YOLO is the use of anchor boxes, which allow the model to predict multiple objects of different shapes and sizes in the same cell. This feature makes YOLO

highly effective for detecting objects in complex scenarios, such as heads in crowded settings.

The YOLO (Lin and Sun(2018))algorithm has evolved significantly over time. The first version, YOLOv1, introduced the concept of single-shot object detection, which made it fast but less effective in detecting small objects or objects close to each other. YOLOv2 (Han et al.(2021)Han, Chang, and Wang)improved on this by adding features like anchor boxes and better training techniques, making it more accurate and versatile. YOLOv3 (Farhadi and Redmon(2018))introduced multi-scale predictions, enabling the model to detect objects of different sizes more effectively. Later versions, such as YOLOv4 and YOLOv5(Lu et al.(2022)Lu, Yu, Gao, Li, Zou, and Qiao), focused on improving speed, accuracy, and training efficiency through advanced techniques like cross-stage partial connections and better data augmentation methods. The most recent versions, like YOLOv10(Guan et al.(2024)Guan, Lin, Lin, Su, Huang, Meng, Liu, and Yan), have pushed the boundaries further by incorporating new architectural advancements and optimizing the model for real-world applications.

This continuous evolution of YOLO(Sudharson et al.(2023)Sudharson, Srinithi, Akshara, Abhirami, Sriharshitha, and Priyanka) has made it a popular choice for head counting and other object detection tasks. Its ability to process images in real-time and handle challenging scenarios, such as occlusions and varying lighting conditions, makes it a reliable tool for researchers and practitioners working in this field.

signed to tackle the challenges of modern object detection, such as handling small-scale objects, overlapping instances, and varying lighting conditions. As shown in Fig 1, YOLOv10 consists of three main components:

2.1 Backbone

The backbone is responsible for feature extraction from the input image. In YOLOv10, it employs advanced convolutional layers with mechanisms like Spatial Pyramid Fast Fusion (SPFF). These enhance the ability to capture features at different scales, making the detection of smaller objects more reliable. This is crucial in crowded scenes where individual heads can be small and partially occluded.

2.2 Neck

The neck focuses on aggregating and refining the features extracted by the backbone. It uses multi-scale fusion strategies to prepare the features for detection. The inclusion of attention-based modules such as C2Spatial Attention (C2PSA) ensures that the network prioritizes the most relevant areas of the image for head detection, even in complex and cluttered environments.

2.3 Head

The head is where bounding boxes and class predictions are generated. YOLOv10's detection head has been optimized to provide high-speed, accurate predictions, leveraging its unique architecture for processing outputs from the neck. This design allows the model to excel in real-time applications where speed and precision are both critical.

These components, when integrated, result in a robust architecture that significantly outperforms its predecessors in detecting small objects like heads in dense crowds. This makes YOLOv10 an ideal candidate for crowd analysis tasks, particularly for head counting.

Moreover, the advancements in YOLOv10's architecture also improve its efficiency on edge devices, ensuring that it maintains real-time processing speeds without compromising on accuracy. This opens up opportunities for deploying the model in scenarios like surveillance and crowd management, where computational resources are often limited.

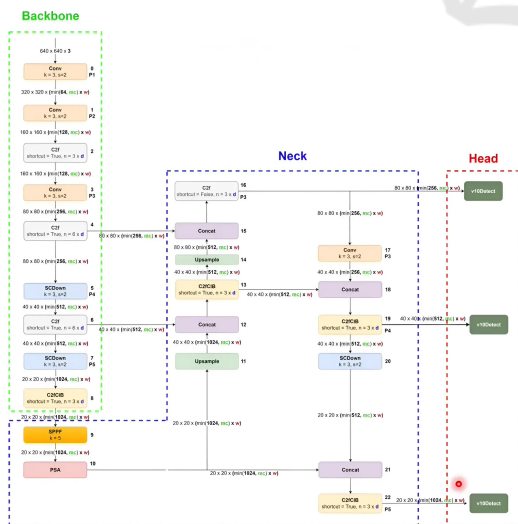


Figure 1: Architecture of the YOLOv10 model, showing the backbone, neck, and head components.

The architecture of YOLOv10 is specifically de-

3 PROPOSED METHODOLOGY

The proposed work focuses on developing a real-time head counting system using YOLOv10, optimized for crowded environments. The workflow includes dataset preparation, fine-tuning a pre-trained YOLO model, and validating predictions with metrics. This approach ensures accurate and efficient detection of heads in diverse and dynamic crowd scenarios.

3.1 Model Selection and Motivation

The model selection for this study is driven by the need for an efficient and accurate solution to crowd counting in dense environments. YOLOv10 was chosen because of its advanced features, such as Spatial Pyramid Fast Fusion (SPFF) and C2 Spatial Attention (C2PSA), which significantly enhance the model's ability to detect small-scale objects and effectively manage overlapping instances. These features allow the model to focus on individual objects even in cluttered scenes, improving detection accuracy. The combination of speed, accuracy, and flexibility makes YOLOv10 an excellent fit for real-time edge device applications. It is capable of delivering reliable performance across various datasets and scenarios, ensuring robust results in real-world situations and making it the ideal choice for this study.

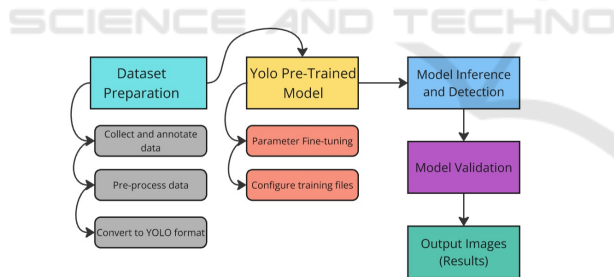


Figure 2: Workflow Diagram Illustrating the Model Training Process, from Dataset Preparation and Pre-trained YOLO Model Fine-tuning and Final Model Validation

The workflow, shown in Fig 2, starts with preparing a dataset by collecting and annotating images with head bounding boxes. The dataset is split into training, testing, and validation sets and converted into a YOLO-compatible format. A pre-trained YOLO model is then fine-tuned for crowd counting tasks. After training, the model predicts head locations in test images, and its accuracy is validated using metrics like mAP@50 and mAP@[50:95]. The process concludes with output images showing detected heads and validation results, demonstrating the model's ef-

fectiveness in handling crowded scenarios.

The YOLOv10 architecture used in this study comprises three main components: Backbone, Neck, and Head, optimized for accurate head counting in crowded settings. The Backbone extracts features using convolutional layers and residual modules, capturing local and global context while reducing spatial dimensions. The Neck enhances feature aggregation through multi-scale fusion using Spatial Pyramid Fast Fusion (SPFF) and C2 Spatial Attention (C2PSA), improving the detection of individuals in various sizes, poses, and orientations. Finally, the Head predicts bounding boxes, class probabilities, and confidence scores via multi-scale detection layers, ensuring precise identification of individuals, even in dynamic and occluded environments.

3.2 Model Training

In terms of the loss function, YOLOv10 employs a combination of classification, objectness, and localization losses to optimize the model for real-time detection. As shown in Equation 1, the total loss function is a weighted sum of these components:

$$L = \lambda_{cls} \cdot L_{cls} + \lambda_{obj} \cdot L_{obj} + \lambda_{loc} \cdot L_{loc} \quad (1)$$

Where: L_{cls} : Classification loss, which measures how well the model identifies objects in the image, L_{obj} : Objectness loss, which checks how confident the model is about detecting objects in the bounding box, L_{loc} : Localization loss, which looks at how accurately the model predicts the position of objects, λ_{cls} : Weight factor for the classification loss, λ_{obj} : Weight factor for the objectness loss, λ_{loc} : Weight factor for the localization loss. These weight factors control the importance of each loss term, helping the model balance its focus during training.

The classification loss is calculated using softmax cross-entropy Equation 2 as follows:

$$L_{cls} = - \sum_{i=1}^C p_i \log(\hat{p}_i) \quad (2)$$

Where: C : Number of classes, p_i : True probability for class i , \hat{p}_i : Predicted probability for class i .

As shown in Equation 3, the objectness loss is computed using the binary cross-entropy between the true label y and the predicted probability \hat{y} .

$$L_{obj} = - [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (3)$$

Where: y : Ground truth objectness score, \hat{y} : Predicted objectness score.

The localization loss, computed using Complete IoU (CIoU) (Equation 4), measures the alignment of predicted bounding boxes with ground truth and takes into account distance, overlap, and aspect ratio differences as follows:

$$\text{CIoU} = 1 - \text{IoU} + \frac{\rho^2(b, b_{\text{gt}})}{c^2} + \alpha v \quad (4)$$

Where: $\rho(b, b_{\text{gt}})$: Euclidean distance between the centers of the predicted and ground truth bounding boxes, c : Diagonal length of the smallest enclosing box around the predicted and ground truth boxes, v : Aspect ratio term, which is typically used to account for the difference in the shape of the predicted and ground truth boxes, b : Predicted bounding box, b_{gt} : Ground truth bounding box.

3.3 Validation and Testing

After training, the YOLOv10 model undergoes validation and testing to ensure robust performance. Validation involves assessing metrics such as mean Average Precision (mAP@50 and mAP@[50:95]) on a subset of data to fine-tune hyperparameters and avoid overfitting. Testing evaluates the model on an unseen dataset with diverse conditions, including varying lighting, crowd densities, and occlusions, using metrics like Precision, Recall, F1-Score, and Inference Time. Visualizations of detected bounding boxes on test images demonstrate the model's capability to identify heads accurately in challenging scenarios. These results confirm YOLOv10's effectiveness and real-time applicability for head counting in crowded environments.

4 RESULT

This section presents the results of the YOLOv10-based head counting system. The model's performance is analyzed through both qualitative and quantitative evaluations. The results include details about the dataset, validation accuracy trends, and detection outputs in sparse and dense crowd scenarios, highlighting the model's effectiveness and reliability.

4.1 Dataset Description

The Crowd Counting Dataset (crowd-counting-dataset-w3o7w) contains a total of 2898 RGB images in JPEG (.jpg) format. These images were captured under diverse conditions, representing crowded scenes with varying densities and perspectives. Each image is annotated with bounding boxes to mark the

locations of individuals in the crowd, providing detailed labeling for training and evaluation. The annotations are stored in a structured format compatible with the YOLO training pipeline, ensuring easy integration for object detection tasks.

4.2 Model accuracy and inference

The model's performance was evaluated using validation accuracy, which was measured by mAP50. As shown in Fig 3, the validation accuracy changes progressively with the number of training epochs. Initially, the accuracy starts at around 23%, then increases steadily, though with some fluctuations, eventually reaching approximately 49% after 40 epochs. This gradual improvement suggests that the model is successfully learning from the data and beginning to perform optimally at this stage.

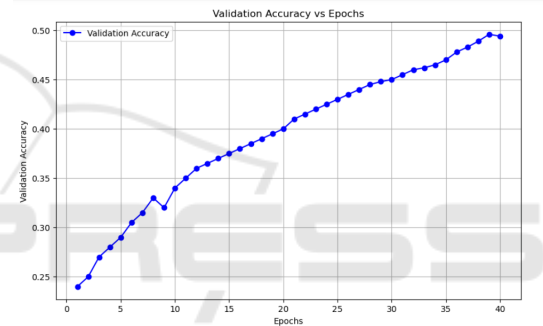


Figure 3: Representing validation Accuracy vs epochs

The increase in accuracy highlights the effectiveness of the YOLOv10 architecture in learning from complex crowd scenarios. The model's advanced feature extraction, enabled by its layers and attention mechanisms, plays a crucial role in addressing challenges like overlapping people, varying lighting conditions, and detecting small objects. The steady rise in accuracy demonstrates how efficiently the model adapts to the data, learning to recognize patterns and improve its performance with each epoch. By the 40th epoch, the model has reached a solid level of competence, indicating that it is well on its way to providing accurate crowd counting in diverse and difficult environments.

Fig 4 demonstrates the results of the object detection process applied to a scenario using YOLOv10 model. The output image displays a group of seven individuals detected within the frame. Each individual is identified with bounding boxes, and their total count is prominently presented as "Number of people detected: 7." This result highlights the accuracy of the model in detecting and quantifying individuals in environments with minimal crowding, showcasing

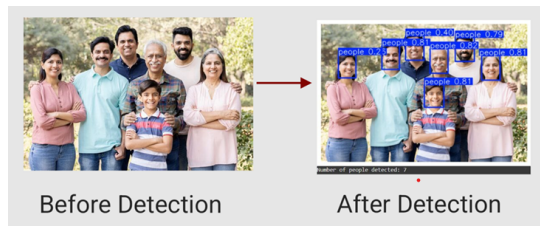


Figure 4: YOLOv10 detection results, identifying 7 individuals in a sparse scene.

its effectiveness in less complex detection scenarios.



Figure 5: YOLOv10 detecting individuals in a moderately dense crowd, with a total count of 69.

Fig 5 showcases the YOLOv10 model's performance in a controlled crowd detection scenario. The "Before Detection" image highlights a moderately dense group, while the "After Detection" output demonstrates accurate identification of individuals with bounding boxes and confidence scores. The total detected count 69, is displayed at the bottom, further confirming YOLOv10's reliability in accurately estimating crowd numbers in less crowded and well-separated environments.

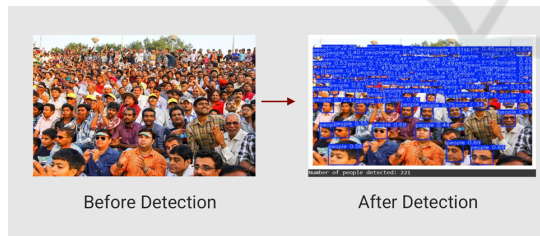


Figure 6: YOLOv10 detecting individuals in a densely packed crowd, with a total count of 221.

Fig 6 highlights the YOLOv10 model's effectiveness in detecting individuals in a densely packed crowd. The "Before Detection" image presents a complex scene with significant occlusion and overlap among individuals. In the "After Detection" output, the model successfully identifies and marks each person with bounding boxes and confidence scores. The total detected count, 221, is displayed at the bottom, demonstrating YOLOv10's robustness in handling challenging crowd scenarios.

The results shown in Fig 5 and Fig 6 demonstrate how well the YOLOv10 model works for counting

people in different situations. Fig 6 shows that it performs well even in dense, crowded scenes, while Fig 5 highlights its accuracy in less crowded settings. The model's ability to count people accurately and provide confidence scores makes it useful for real-world applications, even in difficult conditions. These results prove that YOLOv10 is reliable and scalable for crowd analysis tasks.

5 CONCLUSION AND FUTURE WORK

This research successfully demonstrates that YOLOv10 offers a robust and efficient solution for counting people in dense crowds. Our method achieves notable gains in both accuracy and speed, significantly outperforming traditional methods and other contemporary approaches in detecting and counting individuals, while also requiring less computational power. It effectively addresses common challenges like occlusions, where people are partially hidden behind each other, varied crowd densities ranging from small gatherings to large events, and the detection of small heads, which can often be overlooked in crowded scenes. The model's sophisticated features, including multi-scale detection, which allows it to identify heads of various sizes and distances, and spatial attention, which enables it to focus on relevant parts of the image while ignoring irrelevant details, enable it to handle complex scenes with overlapping individuals, maintaining high performance even in the most challenging situations. These findings underscore YOLOv10's potential for a wide range of real-world applications such as managing crowds at public events and transportation hubs, enhancing event security by providing real-time monitoring of crowd numbers, and monitoring urban areas to understand pedestrian flow and congestion patterns, ultimately leading to safer and more efficient environments, and the better allocation of resources in response to crowd behavior.

Future work will focus on incorporating advanced techniques like multi-scale feature fusion and domain adaptation to further enhance the model's performance across different environments. Specifically, we will explore multi-scale feature fusion, which combines information from various layers of the neural network to enable the model to capture a more comprehensive view of each scene. This should allow the model to recognize objects more accurately, regardless of their size or distance. Furthermore, we will focus on domain adaptation, which involves fine-tuning the model to work effectively in different types of en-

vironments, like various lighting conditions or camera angles. This will improve the model's reliability and ensure that it can accurately count people in any setting. Additionally, efforts will be made to deploy the model on edge devices for real-time crowd counting applications, enabling faster and more efficient monitoring in practical settings, and by this making it easier to process and analyze data directly where it is collected, reducing reliance on remote servers. This will make the system more responsive and efficient and allow for faster response times in critical situations that require immediate analysis.

REFERENCES

- KP Ajitha Gladis, R Srinivasan, T Sugashini, and SP Ananda Raj. Smart-yolo glass: Real-time video based obstacle detection using paddling/paddling sab yolo network 1. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–14.
- Sudhir Sidhaarthan Balamurugan, Sanjay Santhanam, Anudeep Billa, Rahul Aggarwal, and Nayan Varma Alluri. Model proposal for a yolo objection detection algorithm based social distancing detection system. In *2021 International Conference on Computational Intelligence and Computing Applications (IC-CICA)*, pages 1–4. IEEE, 2021.
- Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. Object detection using yolo: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, 82(6):9243–9275, 2023.
- Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. In *Computer vision and pattern recognition*, volume 1804, pages 1–6. Springer Berlin/Heidelberg, Germany, 2018.
- Yanmei Fu, Fengge Wu, and Junsuo Zhao. A research and strategy of objection detection on remote sensing image. In *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 42–47. IEEE, 2018.
- Sitong Guan, Yiming Lin, Guoyu Lin, Peisen Su, Siluo Huang, Xianying Meng, Pingzeng Liu, and Jun Yan. Real-time detection and counting of wheat spikes based on improved yolov10. *Agronomy*, 14(9):1936, 2024.
- Xiaohong Han, Jun Chang, and Kaiyuan Wang. Real-time object detection based on yolo-v2 for tiny vehicle object. *Procedia Computer Science*, 183:61–72, 2021.
- Muhammad Hussain. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11(7):677, 2023.
- Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.
- B Karthika, M Dharssinee, V Reshma, R Venkatesan, and R Sujarani. Object detection using yolo-v8. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–4. IEEE, 2024.
- Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.
- Jia-Ping Lin and Min-Te Sun. A yolo-based traffic counting system. In *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 82–85. IEEE, 2018.
- Yan-Feng Lu, Qian Yu, Jing-Wen Gao, Yi Li, Jun-Cheng Zou, and Hong Qiao. Cross stage partial connections based weighted bi-directional feature pyramid and enhanced spatial transformation network for robust object detection. *Neurocomputing*, 513:70–82, 2022.
- Oluwaseyi Ezekiel Olorunshola, Martins Ekata Irhebhide, and Abraham Eseoghene Ewiekpaefe. A comparative study of yolov5 and yolov7 object detection algorithms. *Journal of Computing and Social Informatics*, 2(1):1–12, 2023.
- Ruchika, Ravindra Kumar Purwar, and Shailesh Verma. Analytical study of yolo and its various versions in crowd counting. In *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021*, pages 975–989. Springer, 2022.
- P Sajitha, Diana A Andrushia, and SS Suni. Multi-class plant leaf disease classification on real-time images using yolo v7. In *International Conference on Image Processing and Capsule Networks*, pages 475–489. Springer, 2023.
- Jun Sang, Zhongyuan Wu, Pei Guo, Haibo Hu, Hong Xiang, Qian Zhang, and Bin Cai. An improved yolov2 for vehicle detection. *Sensors*, 18(12):4272, 2018.
- Yuheng Shi, Naiyan Wang, and Xiaojie Guo. Yolov: Making still image object detectors great at video object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2254–2262, 2023.
- D Sudharson, J Srinithi, S Akshara, K Abhirami, P Sriharshitha, and K Priyanka. Proactive headcount and suspicious activity detection using yolov8. *Procedia Computer Science*, 230:61–69, 2023.
- Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.
- Chien-Yao Wang, Hong-Yuan Mark Liao, et al. Yolov1 to yolov10: the fastest and most accurate real-time object detection systems. *APSIPA Transactions on Signal and Information Processing*, 13(1), 2024.

Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3520–3529, 2021.

Liquan Zhao and Shuaiyang Li. Object detection algorithm based on improved yolov3. *Electronics*, 9(3): 537, 2020.

