IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 October 2023, accepted 19 December 2023, date of publication 26 December 2023,
date of current version 9 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347548

TOPICAL REVIEW

Near-Edge Computing Aware Object Detection: A Review

ARIEF SETYANTO[1], (Member, IEEE), THEOPILUS BAYU SASONGKO[2], (Member, IEEE), MUHAMMAD AINUL FIKRI[2], (Member, IEEE), AND IN KEE KIM[3], (Member, IEEE)

[1]Magister of Informatics, Universitas Amikom Yogyakarta, Sleman 55283, Indonesia
[2]Department of Informatics, Universitas Amikom Yogyakarta, Sleman 55283, Indonesia
[3]School of Computing, University of Georgia, Athens, GA 30602, USA

Corresponding author: Arief Setyanto (arief_s@amikom.ac.id)

This work was supported by the Directorate General of Higher Education, Research, and Technology (DGHERT) of the Ministry of Education, Culture, Research, and Technology Republic Indonesia under Grant No. 181/E5/PG.02.00.PL/2023-0423.19/LL5-INT/AL.04/2023.

ABSTRACT Object detection is a widely applied approach in addressing many real-world computer vision challenges. Despite its importance, object detection is computationally intensive and time-consuming, even with advanced CPU-GPU combinations. With the rise of edge computing and smaller AI accelerators, there is an increasing need to deploy efficient object detection applications on near-edge devices, such as drones and autonomous vehicles. However, these applications often face significant challenges and performance limitations due to restricted computational resources. Traditional object detection methods, e.g., Regional Convolutional Neural Network (RCNN) and You Only Look Once (YOLO), have extensive weight parameters, leading to high demands on memory and computing resources. Therefore, it is important to compress and optimize object detection models by reducing both the size and the number of weight parameters. This review article delves into the current state of object detection methods and simplification strategies, with a focus on deep-learning compression techniques. We investigate various approaches to mitigate these computational challenges, including replacing the regional proposal network (RPN), compressing model backbones, and modifying model heads, specifically for near-edge devices with limited and energy-efficient CPUs and GPUs. While simplifying object detection models is expected to reduce processing time significantly, it can also negatively impact model accuracy. Therefore, we discuss the ongoing challenge of finding the optimal model compression that balances speed while maintaining high accuracy.

INDEX TERMS Compression, edge computing, RCNN, object detection, YOLO.

I. INTRODUCTION

Object detection is an important task in computer vision, responsible for localizing and recognizing specific objects in digital images and videos [1]. This technology has two main goals: to *locate an object* in an image and to *determine the object's class*. These two objectives can be achieved by a unified operation that includes segmenting the image to isolate objects and then classifying each segmented area, thus combining object localization and classification into coa cohesive process. The segmentation of an image into semantically meaningful objects has been a key area of study in computer vision for a long time. Yet, there is still room for improvement in achieving completely accurate segmentation because of the differences between how computers process images and how humans see them. The second aspect of object detection is the recognition of these segments and their categorization into specific object classes. Object detection is closely related to object classification as well as semantic and instance segmentation. It is a computer technology that falls under the broader domain of computer vision and image processing, comprising of the identification and localization of specific objects, such as humans, buildings, or cars, in digital images and videos.

The associate editor coordinating the review of this manuscript and approving it for publication was Christos Anagnostopoulos.

Object detection plays a significant role in computer vision and is renowned for its wide range of applications in various domains like scientific research and industrial production. Prominent examples include multi-category [2], [3], face [4], text [5], pedestrian [6], logo [7], video [8], [9], vehicle [10], and medical image detection [11], [12]. The development of object detection has experienced a significant shift from traditional methods to modern deep learning-based approaches. Historically, object detection techniques before the advent of deep learning relied on manually designed features and heuristic methods for identifying and locating objects in images.

The advent of deep learning, particularly the emergence of Convolutional Neural Networks (CNNs), has presented a major breakthrough in this field. CNNs have shown remarkable accuracy and efficiency in image classification [53], [54], [56], paving the way for adopting CNNs as the backbone architecture of deep learning-based object detection. Within this domain, two prominent approaches have emerged, known as *one-stage* and *two-stage object detection* methods, which have emerged as the most prominent in computer vision.

The modern object detection approaches based on deep learning are distinguished by their underlying architecture and the number of stages in the detection process. One-stage object detection accomplishes the task of detecting and localizing objects in a single step. This method typically employs a dense array of predefined bounding boxes across the entire image and categorizes them into different object classes. Examples of the one-stage methods include YOLO (You Only Look Once) [14] and SSD (Single Shot MultiBox Detector) [15]. These models are designed for real-time performance and are particularly effective in scenarios requiring fast object detection, such as autonomous driving and video surveillance.

On the other hand, as the name of the method indicates, the two-stage method has two phases. Initially, this method generates a set of region proposals to identify potential object locations. These are then refined in a subsequent phase for accurate classification and localization. A widely recognized example of a two-stage object detection framework is Faster RCNN (Regional Convolutional Neural Network) [16]. Due to its computational intensity, the two-stage method achieves higher detection accuracy, making it well-suited for applications that prioritize precision, such as fine-grained object recognition and medical imaging.

Deploying object detection on edge devices, such as Raspberry Pi and Nvidia's Jetson boards, introduces significant challenges due to their limited resources [33], [34], [35]. These challenges include the limitation of power consumption, GPU performance and capacity, CPU cores and speed, as well as the size of RAM available on these devices. Although there are ways of augmenting edge computing power with cloud resources (e.g., task offloading), our study is focused exclusively on object detection computations performed on edge devices without relying on cloud resources. This approach is crucial for maximizing the advantages of edge computing, including real-time processing capabilities and the protection of user privacy.

Consider a scenario where a smart surveillance system is deployed in a public space. If object detection is conducted directly on edge cameras, they can instantly detect and send alerts for any suspicious activities, providing real-time responsiveness. In contrast, if the video stream is transmitted to the cloud for processing, data transmission will be delayed; retrieving results could result in missed or delayed alerts, which may compromise security [105]. Additionally, transmitting sensitive images to the cloud for analysis brings up serious privacy concerns, including issues about data ownership and confidentiality [106]. In this scenario, computation exclusively on edge devices becomes a strategic choice. It ensures immediate response, minimizes latency, and bolsters data privacy. This method can be considered a more appropriate and effective solution, specifically tailored to the application's unique requirements.

Performing object detection directly on edge devices, rather than offloading these tasks to the cloud, can offer several advantages. These benefits include:

1) **Real-Time Responsiveness:** Object detection at the edge allows for real-time inference, eliminating the communication delays associated with cloud processing. This is particularly important for time-sensitive applications like autonomous vehicles and surveillance systems, where immediate detection and response become essential requirements.
2) **Reduced Latency:** Direct object detection on edge devices is especially beneficial for applications requiring low-latency computation, such as augmented reality (AR). In AR, any noticeable delay can negatively impact the user experience. Edge computing avoids the need to transmit data to remote cloud servers for processing, thereby significantly reducing delays.
3) **Privacy and Data Security:** Edge computing offers an effective solution by facilitating the processing of data locally, right where the data is generated, thereby eliminating the necessity for extensive data transmission. This approach particularly protects sensitive data, like images or videos, as it avoids sending this information to remote cloud servers. As a result, it substantially lowers the risk of data breaches, thereby ensuring enhanced privacy and security.
4) **Bandwidth Efficiency:** By processing data locally, edge devices with limited bandwidth can conserve network resources. This is particularly important in scenarios involving multiple edge devices. If each device were to send data to the cloud, it could quickly result in network congestion. Local processing on edge devices helps mitigate this issue.

However, performing object detection solely on edge devices also presents several challenges:

1) **Limited Computational Resources:** Edge devices often have limited computational power and memory

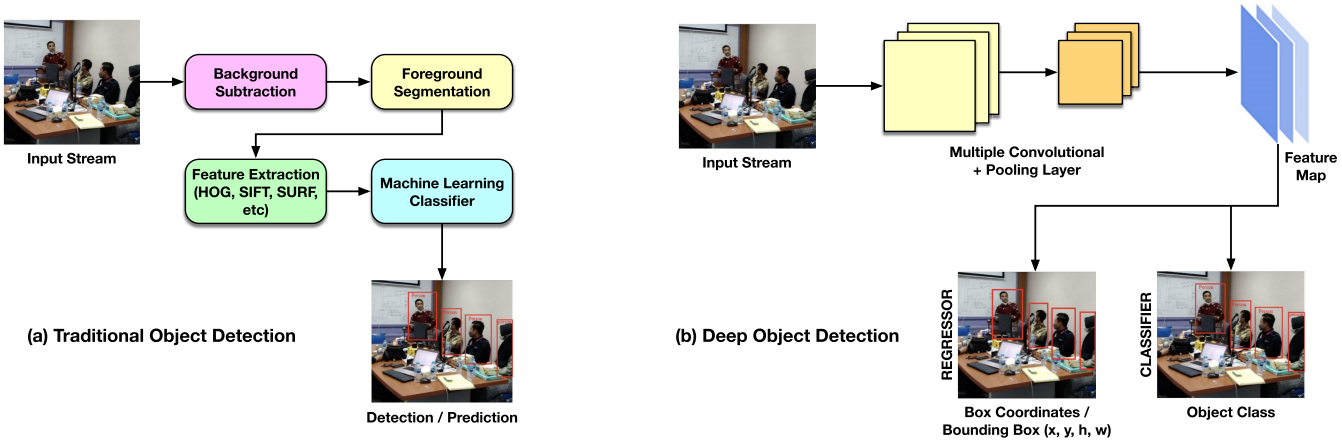

FIGURE 1. Traditional vs. Deep object detection approahces.

compared to cloud servers. In many cases, executing complex object detection models on these devices efficiently necessitates optimization or model simplification.

2) **Model Size and Complexity:** Deep learning models for object detection are generally large and complex, posing a challenge to accommodate them within the limited memory of edge devices. To address this issue while still preserving accuracy, it is important to employ model compression techniques. These methods aim to reduce the model size without significantly affecting its performance.
3) **Energy Consumption:** Computation on edge devices can lead to increased energy consumption, resulting in reduced battery life for battery-powered devices. To mitigate this problem, implementing energy-efficient methods and using hardware accelerators, such as GPUs or Edge TPUs, is essential for better energy management.

To overcome the above limitations in object detection tasks at the edge, compression techniques can be effective solutions. These methods aim to reduce the size and complexity of deep learning models while preserving their original models' accuracy. By doing so, the compression techniques can facilitate deploying object detection models on resource-constrained devices like mobile phones and embedded systems, where managing limited resources is important.

More specifically, complex object detection models can be optimized for edge devices by employing a range of model compression strategies, such as network pruning [17], [19], quantization [20], [23], knowledge distillation [24], [27], and architecture design optimization [28], [29], [30]. Network pruning aims to remove redundant or less important parameters, while quantization reduces the bit precision of model weights. Knowledge distillation transfers knowledge from a larger pre-trained model (e.g., a teacher model) to a smaller one (e.g., a student model). Additionally, architectural design optimization, as observed in models like Fast-YOLO [28], Pelee [29], and Efficient-Net [30] aims to develop compact yet efficient network architectures specifically tailored for object detection tasks.

These compression and optimization techniques provide two key advantages. First, the compression techniques reduce the memory footprint and computational demands, enabling efficient inference on edge devices with limited resources. Second, they enable faster inference and lower power consumption, which is particularly beneficial for real-time applications.

This review article provides a comprehensive overview of state-of-the-art advancements in object detection models and compression techniques. We evaluate their efficacy regarding detection accuracy, model size reduction, and inference speed. Additionally, the article discusses the challenges in this domain and explores potential directions for future research. The insights gained from this work aim to significantly contribute to developing more efficient and practical object detection systems, especially for resource-constrained edge environments. Moreover, these methods offer a viable solution to the limitations of traditional models, enabling the implementation of efficient and accurate detection systems on edge devices. The continued exploration and enhancement of these techniques are essential for expanding object detection applications in various fields, such as robotics, surveillance, autonomous vehicles, and smart devices.

The rest of this paper is organized as follows: Section II discusses the motivation and contribution of this work with a focus on the growing demand for efficient and lightweight object detection models, particularly for deployment in edge computing environments. In Section III, we provide a comprehensive review of the evolution of object detection, transitioning from traditional methods to those based on deep learning. Section IV reviews deep learning compression methods, including knowledge distillation, pruning, and quantization, and their application in object detection. Section V describes approaches for simplifying object detection, such as replacing the backbone, streamlining the Region Proposal Network (RPN), and refining the model head. Section VI discusses the use of object detection in

edge devices, focusing on aspects like the capacities of these devices and the backbone of object detection. Finally, Section VII concludes this paper and outlines future research directions.

II. MOTIVATION AND CONTRIBUTION

Compressing object detection is carried out to address the increasing demand for efficient and lightweight models suitable for deployment in edge computing environments. This research is motivated by the necessity to refine object detection techniques for smooth operation on resource-constrained edge devices. These devices frequently face constraints in processing power and memory, making it challenging to run complex deep-learning algorithms effectively.

The primary objective of compression object detection is to reduce computational complexity and model size while maintaining acceptable accuracy. By achieving model compression, object detection can be deployed on edge devices without compromising performance. This optimization is crucial for real-time applications where low latency and rapid response are essential. Several solutions can be applied to address the challenges associated with object detection on edge devices, including limited computational resources, model size, complexity, and energy consumption.

One viable approach is the application of model compression techniques, such as knowledge distillation or pruning, aimed at reducing the size of the object detection model without significantly sacrificing its performance. Quantization, which necessitates reducing the precision of the model weights, serves as an effective approach to minimize memory requirements and improve inference speed. Another solution is to explore lightweight backbone architectures specifically designed for edge devices. These specialized architectures are designed to have fewer parameters and computations while maintaining acceptable levels of accuracy. Some significant examples include MobileNet [31], SqueezeNet [32], and Efficient- Det [30], all recognized for their efficiency on resource-constrained devices. To further enhance energy efficiency, hardware acceleration can be leveraged through the use of specialized processors or accelerators such as Tensor Processing Units (TPUs) [33] and Field Programmable Gate Arrays (FPGAs) [36].

There are some dedicated hardware components developed to accelerate neural network computations, enabling swifter and more energy-efficient object detection on edge devices. For example, Intel has introduced the OpenVINO Toolkit, compatible with a wide range of Intel chips, including CPUs, GPUs, FPGAs, and vision processing units [37]. Similarly, Nvidia has introduced the EGX platform, offering support for its various hardware, from the lightweight Jetson Nanos to the powerful T4 servers [38].

In this research, a comprehensive review of compression methods in object detection for edge devices was provided and the challenges encountered were analyzed based on state-of-the-art literature. The main contributions of this research are as follows:

1) The study concentrated on exploring how compression techniques can be applied effectively in edge devices.
2) The study involved an in-depth examination of widely used techniques, focusing on the compression and replacement of backbone networks as well as the simplification of the Region Proposal Network (RPN) head.

Therefore, a comprehensive understanding of these methods and their effects is essential for effectively deploying object detection systems on edge devices.

III. OBJECT DETECTION METHODS

Object detection is classified into two main methods: traditional and deep-learning-based approaches. Traditional approaches, steeped in a long history, are typically used for detecting specific objects like faces or human bodies. It extracts features from the image using specialized filters and systematically scans them across the image.

This **conventional approach**, steeped in a long history, is typically used for detecting specific objects like faces or human bodies. It extracts features from the image using specialized filters and systematically scans them across the image. The **deep-learning-based approach** utilizes CNNs for feature extraction. Certain deep learning methods involve two steps: 1) initial pre-computing region proposals and 2) classifying each region into specific object categories. Alternatively, other approaches aim to address challenges in the two-stage process by integrating region proposal generation and object localization inference into a single computational step. Additionally, the availability of public datasets is crucial, as it facilitates the effective training and testing of these methods.

A. TRADITIONAL APPROACH (NON-DEEP LEARNING APPROACH)

Object detection, as previously mentioned, is an important area of computer vision research, with a particular emphasis on detecting human forms like faces [45]. Haar cascade classifiers have been recognized as a significant method to detect and localize faces and other objects effectively when trained with ample positive and negative images. This method involves convolution operations between the input image and a series of predefined filters. A key characteristic of Haar cascade classifiers is their notable speed and efficiency, which has led to their widespread use in various applications. Another significant approach focused on face detection employs the Histogram of Oriented Gradients (HOG). Unlike Haar cascades, HOG assesses gradients at individual points or pixel clusters in the image. This approach calculates the gradient magnitude and direction at each point, compiling these into a histogram of gradients. These features are then utilized to identify specific objects within the image, demonstrating the method's distinctiveness and effectiveness in object recognition tasks.

TABLE 1. Comparison of existing survey object detection in the edge.

Authors	Years	Strength	Contributions	Limitations
Huang et al. [39]	2021	The review discusses the use of a backbone CNN on Raspberry Pi.	The section provided summaries of various backbone models used on Raspberry Pi.	The evaluation focused solely on assessing the efficiency of the new CNN-RIS model in facial expression detection.
Wang et al. [40]	2020	The survey provided a comprehensive presentation and exploration of numerous relevant scenarios and fundamental enabling techniques for both edge intelligence and intelligent edge.	The objective was to improve the performance of deep learning training and inference. It could be achieved by considering various constraints, including networking, communication, computing power, and energy consumption.	This review did not include the optimization of all components within object detection architectures, including both one and two-stage methods.
Liu et al. [41]	2019	The presented article introduced prominent systems and open-source projects in the field of edge computing. It also discussed various strategies to enhance energy efficiency for improved performance and explored technologies for deploying deep learning methods at the edge.	This review mainly concentrated on edge computing systems and tools, exploring innovations in architecture, programming models, and applications within the field of edge computing.	This review did not specifically cover object detection at the edge.
Kang et al. [42]	2022	The review showed several advanced devices, including the Google Coral Dev Board Mini, NVidia Jetson Nano, and NVidia Jetson Xavier, feature a range of accelerator architectures serving as coprocessors. These accelerators had the ability to significantly enhance the performance of AI applications.	The capabilities of each device were assessed by using a range of object detection models, including YOLOv4-Tiny and SSD MobileNet V2. These evaluations aimed to measure their detection accuracy, performance latency, and energy efficiency.	This review exclusively assessed the object detection capabilities of YOLOv4-Tiny and SSD MobileNet V2 on edge devices such as the Google Coral Dev Board Mini, NVIDIA Jetson Nano, and NVIDIA Jetson Xavier. However, it failed to offer solutions for the issues encountered during the evaluation process.
Zou et al. [13]	2023	The review showed the survey of 20 years object detection methods cover classic approach such as haar cascade to the recent deep learning based object detection such as RCNN, YOLO and SSD	Their contribution are defining the development of object detection in historical context during the last 20 years and its capability improvements	Although they cover a topic of model compression, their discussion still limited in post training quantization and prunning

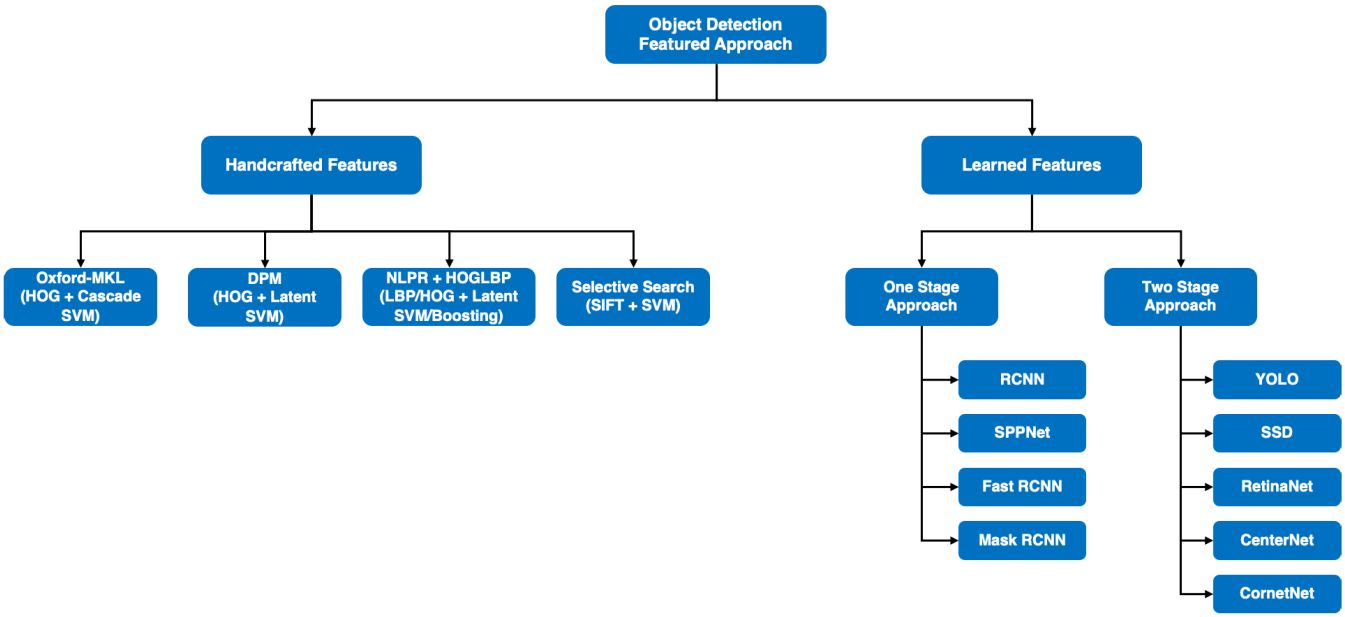

FIGURE 2. Object detection features approach.

Oxford-MKL (Multiple Kernels for Object Detection) [46] represents a significant advancement in object detection. Its main goal is to create an advanced object category detector using a state-of-the-art classifier. This classifier carefully examines all potential sub-windows within an image for the target object's presence. To facilitate this process, Oxford-MKL employed multiple kernel learning [47], enabling the optimal fusion of exponential kernel, each focusing on different feature channels, such as edge distribution, dense and sparse visual words, and feature descriptors at multiple spatial levels. Given the time constraints in testing the robust classifier on each image sub-window, Oxford-MKL introduces a three-stage classifier. This classifier combines linear, quasi-linear, and non-linear kernel Support Vector

Machines (SVMs) for efficient processing. The Oxford-MKL Framework enhances the discriminatory capabilities of the kernel by increasing the non-linearity, thereby introducing a higher level of computational complexity.

Deformable Parts Model (DPM) [48] is another important approach in object detection. The method not only outperformed the previously popular HOG but also secured first place in the 2009 Pascal VOC challenge [43]. DPM led to a paradigm in the way object detection was approached. Instead of treating objects as undivided wholes, it adopted a divide-and-conquer strategy. DPM uniquely detects individual parts of an object and then intelligently combines them to form a complete piece. The method proved particularly effective when applied to articulate objects exhibiting varying poses, like the human body composed of the head, arms, legs, and torso. Each part was assigned a specific model, and the detection process was systematically applied to all parts, thereby eliminating unlikely combinations and yielding highly accurate results. DPM-based models, such as those proposed in [49] and [50], showcased significant performance prior to the advent of deep learning. While traditional methods, such as simple template-based object models, struggled with geometric deformations, and bag-of-words had limitations in precise object localization, DPM successfully addressed these challenges.

Object Detection by Context and Improved HOG-Local Binary Patterns (LBP) [51] capitalizes on the synergy of contextual information, while a combination of HOG and LBP features is further amplified through boosting. The main goal of this method is to enhance the accuracy and robustness of object detection. This is achieved by considering the surrounding context of objects and leveraging the discriminative capabilities of HOG and LBP features. By integrating contextual information, the method delivers improved results when objects vary in appearance and size and are partially obscured. This enhanced approach boosts the discriminative power of the feature representation and effectively manages complex background disturbances. As a result, it significantly contributes to the progression of object detection, overcoming some of the limitations inherent in traditional methods. This advancement underscores the importance of contextual understanding and refined feature representations in achieving higher detection accuracy.

Selective Search (SIFT+SVM) [52] combines the Scale-Invariant Feature Transform (SIFT) method with the SVM classifier. This method aims to efficiently identify potential object locations within an image, simplifying the object recognition process. By using the capabilities of SIFT, known for its dominant performance in extracting stable and invariant features from images, selective search effectively captures the structural and visual information needed for object recognition. SVM classifier, a popular machine learning method, is then used to classify the extracted features and determine whether an object exists in a given location. Selective search achieves this by hierarchically grouping similar image regions based on their resemblance and spatial relationships.

The grouping process enables the method to generate a diverse set of potential object locations, covering various scales and positions within the image. To accomplish this, it combined multiple image segmentation strategies and used a similar measure to guide the grouping process effectively. The selective search method has proven to be highly promising, delivering a balanced trade-off between recall and computational complexity, enabling a more efficient object recognition process. By producing a diverse set of potential object locations, selective search significantly enhances the performance of the SVM classifier, thereby leading to improved accuracy in object recognition.

B. DEEP LEARNING APPROACH: DOUBLE STAGE COMPUTATION

The advent of CNN marked a significant milestone in image classification, notably with the groundbreaking work by AlexNet in 2010 [53], which initially focused on accurately recognizing handwritten characters. Subsequently, a series of breakthroughs have been recorded, resulting in the widespread adoption of CNN architectures such as VGG-16 [54], InceptionNet [55], ResNet [56], XceptionNet [57], etc. These models, often trained on extensive datasets like ImageNet [58], demonstrated exceptional performance, allowing their weight parameters to be fine-tuned for new problems, an approach known as transfer learning. This concept of transfer learning was initially applied in the early stages of deep learning-based object detection. While R-CNN was the first successful object detection method, it suffered from computationally intensive processes, mainly due to the preparation of region proposals.

R-CNN, developed by Girshick et al. [59], was a significant advancement in object detection, bringing the potential of CNN to the forefront of object detection. R-CNN uses a class-agnostic region proposal module that uses selective search to generate approximately 2,000 potential object candidates. These candidates are then passed through a CNN, typically AlexNet [53], integrated with a region proposal selective search [52] to extract a 4,096-dimensional feature vector for each one.

The method utilizes trained class-specific SVMs to assign confidence scores and applies non-maximum suppression (NMS) to eliminate overlapping detections. It also includes a bounding box regressor for precise object localization. The training of R-CNN involves initially pre-training the CNN on a large dataset like ImageNet, followed by fine-tuning it on domain-specific images. This fine-tuning replaces the last fully connected layer with a newly initialized N+1-way classifier, where N represents the number of object classes. Separate SVMs and bounding box regressors are trained for each object class. However, R-CNN faced challenges with slow inference time, approximately 47 seconds per image, and required substantial computational resources [60].

RCNN operates as a detector with four key components: region proposal generation, feature extraction, SVM classification, and bounding-box regression. It uses a selective search for region proposal generation and extracts fixed-length feature vectors from each using CNN. The classification is done using class-specific linear SVMs, while the bounding-box regressor accurately determines object boundaries.

This research highlighted the importance of pre-training CNNs on larger datasets like ImageNet, followed by targeted fine-tuning. During fine-tuning, a (N+1)-way classification layer is initialized and optimized using SGD. The training process involves defining positive and negative examples based on IoU (Intersection Over Union) thresholds, with region proposals classified as positives or negatives depending on whether they fall above or below the threshold. To prevent overfitting, the research included a larger set of positive examples with IoU overlaps between 0.5 and 1, even if they are not exact matches to ground truth instances. Combining CNNs with region proposals, R-CNN significantly improved object detection performance on datasets like PASCAL VOC, outperforming methods reliant on simpler features like HOG.

C. DEEP LEARNING APPROACH: SINGLE STAGE COMPUTATION

Single-stage object detection is an advanced computer vision method designed to swiftly and accurately identify objects in an image in a single stage. Unlike the traditional two-stage method that requires complex region proposal and subsequent object classification, single-stage detectors directly predict object bounding boxes and class probabilities, eliminating the need for an intermediary stage. The streamlined method has gained immense popularity for its real-time performance and high accuracy, making it suitable for various applications, including autonomous vehicles, surveillance systems, and image analysis tasks. In this context, the principles, advancements, and challenges of single-stage object detection, including the innovative architectures and techniques that had contributed to its success in recent years, were analyzed.

YOLO [14] represents a well-received object detection model. Unlike two-stage detectors that approach object detection as a classification problem with multiple components, YOLO adopts a unique method by treating it as a regression-related issue. It directly predicts both bounding box attributes and object classes. In the YOLO framework, the input image is divided into an S×S grid, each responsible for detecting objects whose centers fall within its cell. YOLO predicts multiple bounding boxes per cell, each represented by five elements the center coordinates (x and y), box dimensions (w and h), and a confidence score.

Inspired by the GoogLeNet [55] model for image classification, YOLO is pre-trained on ImageNet data and then augmented with additional convolutional and fully connected layers. During training, each grid cell is optimized to predict a single class to achieve better convergence, but it can predict multiple classes during inference. The model is optimized using a multitask loss function that combines the losses of all predicted components. One significant advantage of YOLO is its real-time performance, surpassing other models in accuracy and speed within the single-stage category. However, it has faced certain limitations, including challenges in accurately localizing small or clustered objects and restrictions on the number per cell. These shortcomings were addressed in later versions of YOLO. As a one-stage object detector, YOLO achieves real-time detection for all images and webcam streams by predicting fewer bounding boxes per image compared to alternative methods. It adopts a unique method by framing object detection as a regression problem, allowing a unified architecture to efficiently extract features from input images to predict bounding boxes and class probabilities directly. The network runs at an impressive 45 frames per second on a Titan X GPU, outperforming both Fast and Faster R-CNNs in speed.

YOLO pipeline divides the input image into an S×S grid, with each cell responsible for detecting an object. Confidence scores are determined by considering the probability of an object contained in a box and the IOU, indicating the accuracy of this containment. Each grid cell predicts B bounding boxes, their associated confidence scores, and conditional class probabilities for C categories. The feature extraction network comprised 24 convolutional layers and two fully connected layers. YOLO experiments on the PASCAL VOC dataset yielded impressive results, achieving a mean average precision (mAP) of 63.4% while running at a high speed of 45 frames per second, outperforming both Fast and Faster R-CNNs. Although YOLO may encounter certain challenges with precise localization, its localization errors are significantly lower compared to other methods. As a result, YOLO is an efficient and accurate object detection model, suitable for real-time applications and achieving impressive results in various scenarios.

SSD [15] is a pioneering single-stage method to object detection. Its primary innovation lies in directly predicting category scores and bounding box offsets for a set of predetermined default bounding boxes distributed across multiple feature maps with varying scales. Each feature map specializes in detecting objects at specific scales, achieved by strategically spacing the default bounding boxes across a range of layers. For every default box, SSD predicts both confidence scores and offsets pertaining to all object categories.

To achieve real-time processing speed while maintaining accuracy comparable to two-stage detectors such as Faster R-CNN, SSD incorporates additional auxiliary convolution layers into the VGG-16 architecture. During training, ground truth boxes are matched with the most suitable default ones using the Jaccard overlap, and the network is fine-tuned through a weighted sum loss function and a process called hard negative mining. Despite its impressive performance, SSD tends to encounter certain challenges with respect to detecting small objects. These issues were subsequently

addressed by adopting more advanced backbone architectures, such as ResNet, and implementing minor adjustments to enhance its capabilities.

D. DATASET

Labeled datasets play a significant role in supervised machine learning, serving as essential resources. There are several widely used public datasets for object detection methods such as PASCAL VOC [43], Microsoft COCO [44], VISUAL GENOME, NUScenes, wildfire smoke dataset, etc. In addition to datasets containing object bounding boxes, general datasets with a single label for each image, such as ImageNet, are also crucial for training CNN and building robust models. The availability of pre-trained CNN models is of significant importance for deep learning methods.

IV. DEEP LEARNING COMPRESSION METHODS

Deep learning methods are known for their large size and considerable computational complexity and requirements, posing a significant challenge for deployment on devices with limited resources. To tackle this, various compression techniques have been developed, such as knowledge distillation, pruning, and quantization, aiming at reducing the size and complexity of deep learning models for easier implementation on resource-constrained devices.

A. KNOWLEDGE DISTILLATION

Knowledge distillation is a method aimed at training a compact model by transferring knowledge from a larger, more complex one. The method applied a soft target distribution during training, generated by a cumbersome model with a high temperature of softmax operator, for each instance in the transfer set. Temperature is a variable to set the softmax operator. Larger temperatures lead to generating a softer distribution of pseudo-probabilities among the output classes. Initially, the distilled model is trained with the same high temperature as the source model, but it eventually transitions to a temperature of 1 after the training process. For example, in [61], this research employs MnasNet (1.0x), which is not compact enough to operate on a less powerful device. However, the reduced version of MnasNet (0.5x - student version) can achieve a speed comparable to MobileNet v2 on such a device, albeit at the cost of accuracy. By leveraging the knowledge gained from an ensemble of models or a highly regularized one, the distilled is effectively compressed into a smaller, more manageable form suitable for deployment. Figure 4 illustrates the knowledge distillation concept.

In 2020, Matsubara et al. [61] proposed the Head Network Distillation method to compress the initial layers of DNN. The primary goal was to enable efficient classification of complex images with lower computational demands. By applying the concept of knowledge distillation, Matsubara et al. successfully created a smaller student model capable of matching the performance of the larger teacher model. The results emphasized the effectiveness of the method, as it allowed for aggressive model compression, reduced inference time, and improved accuracy. Li et al. [62] conducted research on knowledge distillation in the field of object detection. The loss values between the teacher and student models were compared, and it was reported that this technique effectively improves performance while reducing the size of the models. This is in line with previous research by [61], further confirming its effectiveness in optimizing these models.

Kang et al. [63], conducted similar research, by introducing a conditional framework. This innovative method enabled knowledge distillation on a per-instance basis, focusing on object classification and localization. This technique facilitated the transfer of specific object-related information to the student model, resulting in improved detection performance. This research reflects ongoing efforts to advance the procedure's sophistication and adaptability in object detection.

Chawla et al. [64], conducted similar research on data-free detection models. It used the Deep Inversion for Object Detection (DIODE) technique, which facilitated knowledge distillation without the need for training datasets. By using DIODE, the effectiveness of this technique was enhanced without training data. This research exemplified efforts to overcome the limitations posed by training data in respect to knowledge distillation.

Bharadhwaj et al. [65] designed the Detect-Track-Count (DTC) framework, which focused on efficiently counting vehicles on edge devices. DTC used ensembled knowledge distillation to improve the detection accuracy of the Tiny YOLO model. This research also leveraged this procedure in order to enhance object detection performance.

Tao et al. [66] proposed an efficient and strong cloud-based machine learning framework that used knowledge distillation in 2023. The Neural Manifold Distillation (NMD) method was used to simplify the complexity of the deep learning method without compromising the performance. The designed framework enabled the use of lighter and more efficient models within a cloud-based environment, thereby enhancing computational efficiency in resource-constrained systems. This research is consistent with previous ones, contributing to the ongoing efforts to optimize knowledge distillation across various contexts and environments.

B. PRUNING

Pruning is a commonly adopted technique for compressing DNN, with the objective of reducing both the model size and computational complexity. This was realized by removing redundant or insignificant connections, weights, or neurons from the network while preserving the entire performance, as shown in figure 5. For example, a study conducted by Liang et al [71]. demonstrates that Edge Yolo, a pruned model from YOLOv4, with a size of 25.27MB, achieves a mAP of 47.3%, while only accounting for 10.2% of the original YOLOv4 (245.78MB), as opposed to its original mAP of 65.7%. This research proves that pruning methods have significant potential in reducing model size with minimal

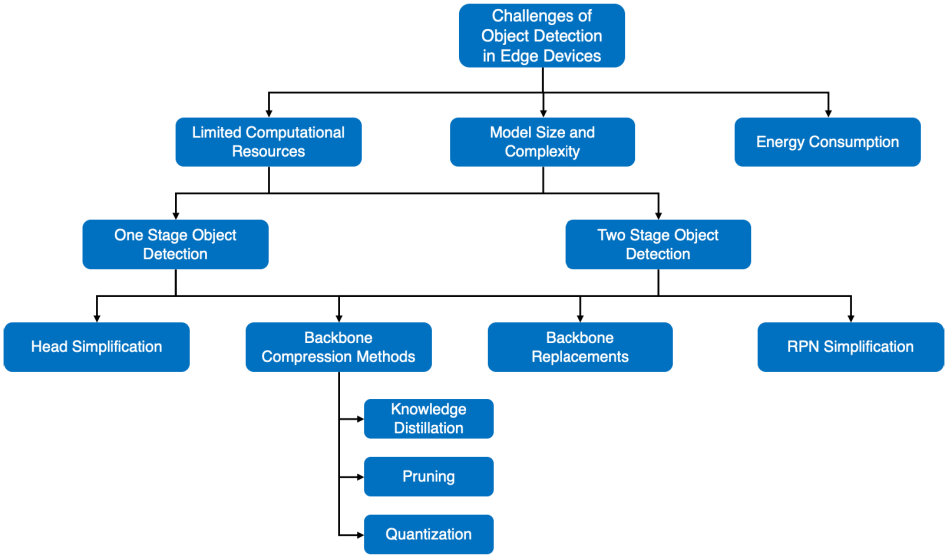

FIGURE 3. Challenges of object detection in edge devices.

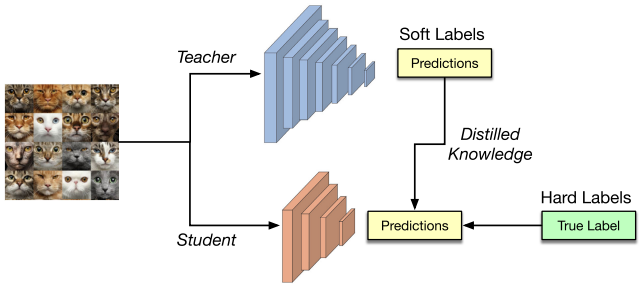

FIGURE 4. Knowledge distillation.

performance deviation. Several of the listed research focused on various pruning methods in DNN, each presenting a unique method and addressing specific limitations in the field.

Li et al. [67] designed DeepCham, an adaptive deep learning method designed for mobile object recognition, leveraging edge computing. This pruning method selectively removes unimportant connections and neurons, resulting in model compression and improved inference efficiency. However, the research lacks a detailed analysis of the limitations or potential trade-offs associated with the pruning process. Han et al. [68] addressed this limitation using Deep Compression, a comprehensive framework that combined pruning with trained quantization and Huffman coding. The proposed method achieved substantial model compression while minimizing accuracy loss. One of the challenges in pruning lies in determining an optimal threshold that strikes a balance between compression and performance.

Hinami and Satoh [69] introduced a novel method for object detection using a large-scale R-CNN combined with classifier adaptive quantization. This innovative method combined two significant methods, e.g., pruning and quantization, which reduce the model size by eliminating redundant parameters and decreasing the precision of weights and activations. The combined method balances accuracy and computational efficiency through adaptive quantization. However, the research did not explicitly discuss the potential limitations or drawbacks of applying the pruning method.

Nguyen et al. [70] conducted research aimed at investigating the effects of weight pruning on YOLO CNN in the context of object detection. A high-throughput and power-efficient FPGA (Field-Programmable Gate Array) implementation of YOLO CNN was created with the goal of reducing model size and improving inference speed through weight pruning techniques. The primary achievement was significant model compression without a substantial loss in detection accuracy. However, the pruning process impacted fine-grained object detection, which was perceived as an important consideration.

Liang et al. [71] conducted research introducing Edge YOLO, an intelligent real-time object detection system designed for autonomous vehicles. The method is based on edge-cloud cooperation to optimize system performance. To achieve this, the pruning method was used to reduce the size of the YOLOv4 backbone network, thereby enhancing the speed and efficiency. A balance was successfully reached between accuracy and speed in AI scenarios on edge computing platforms by strategically reconstructing the backbone layers and implementing channel pruning. However, the research lacked comparative analysis with other state-of-the-art object detection systems and failed to provide a detailed analysis of the trade-off between accuracy and energy consumption.

Lastly, Liberatori et al. contributed to this field by conducting research focused on face mask detection using YOLOv4, a popular object detection model [72]. The research mainly investigated the effectiveness of pruning and quantization methods in optimizing this model for deployment on low-end devices. Two distinct strategies were applied: one-shot pruning with fine-tuning and an iterative one with learning rate rewind, followed by fine-tuning. The

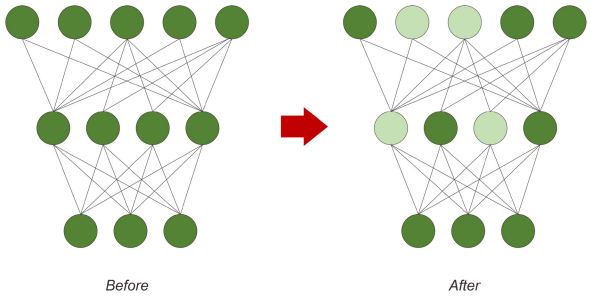

FIGURE 5. Pruning in neural networks.

experimental results showed significant improvements in performance, particularly in terms of frames per second (fps). However, there exists a trade-off between reducing the model size and maintaining mAP levels. One limitation of the research was the absence of a detailed explanation regarding the specific parts of YOLOv4 that underwent pruning and quantization, which are perceived as valuable aspects for a comprehensive understanding of the proposed method.

Pruning offers an effective means of compressing DNN while maintaining its performance. This research investigated various methods, including selective pruning, the combination of pruning and quantization, and pruning designed for specific tasks, namely object detection. Further research is needed to comprehensively investigate the limitations, potential trade-offs, and optimization strategies associated with the pruning method in DNN.

C. QUANTIZATION

Quantization has been perceived as a potential technique for optimizing neural networks in recent years, enabling efficient inference and reducing model size. Like the research conducted by Li et al. [21], they demonstrated that a 4-bit model can perform very closely to the 32-bit floating-point version, even on mobile-friendly networks. Their model, the 4-bit RetinaNet detector with MobileNetV2 backbone, experiences only a 2.0% mAP loss compared to its full-precision baseline. The research in the provided list focuses on various aspects, ranging from adaptive quantization to fully quantized networks. This section analyzed the listed research and their valuable contributions to the quantization field.

Hinami and Satoh [69] proposed a novel approach called large-scale R-CNN with classifier adaptive quantization. This investigation focused on developing object detection methods at a larger scale with limited resources. The evaluation introduced the concept of classifier adaptive quantization (CAQ), which necessitated optimizing the bounding box search process using inverted index and vector quantization residual (RVQ) methods. By leveraging these methods, the goal was to accelerate the object detection process and enhance classification precision. However, the research mainly evaluated the proposed method within the R-CNN framework and did not extensively discuss the potential limitations or trade-offs associated with the adaptive quantization method used.

In 2018, Jacob et al. introduced a novel method titled Quantization and Training of Neural Networks for Efficient Integer Arithmetic-Only Inference [20]. The primary objective was to achieve efficient neural network inference using integer arithmetic, specifically on hardware platforms capable of integer operations. Experiments comprising the quantization of weights and activations into 8-bit integers were conducted while maintaining model accuracy through fine-tuning. However, the research did not thoroughly explore the performance of the proposed scheme on platforms other than MobileNets, and the impact of integer quantization on other neural network types was not discussed in detail. Li et al. further introduced a fully quantized network (FQN) designed for object detection [73]. The goal was to enhance computational efficiency by using low-bit arithmetic accelerated by dedicated hardware. FQN necessitates quantizing network weights and activations using low-bit fixed-point arithmetic. Fine-tuning techniques were used to optimize these quantized weights and activations. However, the uniform nature of the quantization method and the absence of specific adjustments for different networks could limit its optimality. The method was evaluated using the COCO dataset, with particular attention to the computational costs associated with fine-tuning.

Ding et al. proposed the REQ-YOLO framework, focusing on object detection for FPGAs [22]. The block-circulant matrix method was used, while heterogeneous weight quantization was introduced through the Alternative Direction Method of Multiplier (ADMM). The method significantly improved weight compression and storage efficiency by influencing the unique characteristics of block-circulant matrices. However, the evaluation was limited to YOLO and the framework's performance in other models was not addressed. The research lacked comparisons with state-of-the-art quantization methods and did not provide a detailed analysis of the trade-offs between compression and model accuracy.

Gholami et al. conducted a thorough survey on quantization methods for improving the efficiency of neural network inference [74]. The survey explored various aspects of quantization, including symmetric and asymmetric, static and dynamic, granularity, as well as uniform and non-uniform quantization. It emphasized the potential benefits of applying quantization techniques such as INT8 and INT4, which could substantially enhance inference speed while minimizing the loss of accuracy. Furthermore, the challenges inherent in quantization were emphasized, particularly the need to strike a delicate balance between compressing the model and preserving its accuracy.

The earlier discussed research investigated different methods and advancements in neural network quantization, from adaptive quantization to fully quantized networks. These procedures represent a collective effort by preliminary research to optimize network performance and efficiency. Meanwhile, it is clear that there remains a need for further research to effectively address the limitations and challenges

associated with quantization. Investigating its applicability across different network architectures and hardware platforms is an avenue that requires further exploration.

V. OBJECT DETECTION METHODS COMPRESSION

Efficiently compressing object detection models is essential to make them suitable for limited-resource devices. Typically, native object detection methods demand significant memory for model storage and are computationally intensive due to many arithmetic operations. Simplifying these methods becomes crucial to minimize memory usage and computational requirements. Various significant efforts have been proposed to achieve this, including techniques such as backbone replacement, simplification of region proposal network (RPN), and streamlining the model head.

A. BACKBONE REPLACEMENT

In recent years, research has made significant progress in developing efficient backbone architectures for object detection, focusing on improving speed, reducing energy consumption, and accuracy. The collective efforts are poised to revolutionize the field by enabling real-time and resource-efficient object detection applications. Numerous research contributed to this advancement by introducing innovative backbone architectures, each designed to tackle specific challenges and offer novel approaches to object detection.

Lee et al. introduced VOVnet, a significant backbone network designed to improve energy and GPU computation efficiencies while preserving the benefits of DenseNet [75]. This approach necessitated using One-Shot Aggregation (OSA) to merge intermediate features simultaneously, effectively tackling the drawbacks associated with dense connections and reducing energy consumption. VOVnet came in both lightweight and large-scale versions, which were applied to both one and two-stage object detectors. Comparative experiments showed VOVnet's superiority over DenseNet and ResNet, particularly in terms of speed and energy efficiency for real-time object detection. However, its evaluation could have been strengthened by providing a detailed analysis of the trade-offs between accuracy, speed, and energy consumption. A more comprehensive comparison with other advanced object detection architectures would have enhanced its practical relevance. Yukang Chen et al. introduced DetNAS, a framework designed to streamline Neural Architecture Search (NAS) specifically for object detection [76]. It used a one-shot supernet technique to efficiently explore all potential architectures in the search space. This innovative approach led to the development of an architecture that not only surpassed manually crafted networks but also reduced computational complexity, particularly in terms of FLOPs (Floating Point Operations Per Second). As a result, DetNAS provided a valuable advancement in the automated design of efficient backbones for object detection. However, the research did not explicitly mention the limitations of the proposed framework or the scope of its generalization to other detectors and datasets.

Gao et al. proposed a novel architecture known as Res2Net, designed to significantly enhance the multi-scale representation abilities of CNN [77]. This improvement was achieved by introducing hierarchical residual-like connections, effectively expanding receptive fields, and consequently enhancing performance across diverse computer vision tasks. The research introduced both the innovative Res2Net block and a comprehensive description of the entire Res2Net architecture, as well as its compatibility and ease of integration with other neural network modules, including cardinality and squeeze and excitation (SE) blocks. Despite showing competitive results on datasets such as CIFAR-100 and ImageNet, the research had significant limitations. It lacked a thorough analysis of critical factors such as model size reduction and computational cost and failed to include comparisons with the latest neural network modules: EfficientNet and ResNet. These aspects are essential for determining the practicality and adaptability of Res2Net in different resource-constrained scenarios.

Tan et al. addressed the challenge of scalable and efficient object detection by introducing EfficientDet [30]. Several significant optimizations were proposed, including the weighted bidirectional feature pyramid network (BiFPN) for effective multi-scale feature fusion and a compound scaling method. By utilizing EfficientNets, which were pre-trained on the ImageNet dataset, as its backbone architectures, EfficientDet achieved leading performance on the COCO dataset. A series of extensive experiments were conducted to analyze the complex adjustments between model size, computational cost, and accuracy across various resource constraints. This research highlighted EfficientDet as an efficient and scalable architecture for object detection. However, it does not specifically delve into the limitations of this method, nor does it explore how various hyperparameters might affect the model's performance.

Chen et al. [78] addressed the need for lightweight one-stage object detection solutions specifically designed to operate efficiently on CPU-only devices by introducing RefineDetLite. An innovation that cleverly combined the Res2NetLite backbone with the RefineDet module, designed to handle CPU constraints effectively. To validate this approach, a series of experiments were conducted using the MSCOCO dataset. The results showed the ability of RefineDetLite to achieve competitive performance while remaining optimized for CPU-only devices. The research showcased impressive results, outperforming existing methods in terms of efficiency. However, it was important to emphasize that the evaluation was limited to the MSCOCO dataset. Further investigation is required to assess how well the model generalizes to other datasets and tasks. To support the research findings, a direct comparison with GPU-based methods and addressing challenges related to object occlusion would be beneficial.

Hong and Song proposed ResNeXt101S, an improved model for deep object detection backbones, particularly focusing on feature layers [79]. The model aimed to enhance

the quality of features in deeper layers while ensuring compatibility with varying image resolutions. A series of experiments were conducted on the customized benchmark datasets to assess the performance of the model under different image scales. The results indicated that ResNeXt101S demonstrated promising accuracy and adaptability across different resolutions. However, the use of customized benchmark datasets for evaluation brings into question the model's generalizability to real-world scenarios. To provide a more comprehensive understanding of ResNeXt101S capabilities, further evaluation and direct comparisons with state-of-the-art models on a broader range of datasets would be valuable.

Li et al. explored the application of a plain Vision Transformer (ViT) as a backbone for object detection, employing a fine-tuned ViT specifically for this task, thereby avoiding the need for hierarchical backbone redesign [80]. It was found that plain ViT backbones, pre-trained as Masked Autoencoders, were capable of competing effectively with existing methods that rely on more complex hierarchical backbone structures. They introduced ViT-Det, a detector achieving a notable 61.3 AP box score on the COCO dataset, utilizing only ImageNet-1K for pre-training. This approach underscored the feasibility of using general-purpose backbones without task-specific designs, highlighting the importance of separating pre-training from fine-tuning phases. While the method showed considerable promise, further research is necessary to fully evaluate its effectiveness across a wider range of object detection tasks and datasets.

Wang et al. [81] proposed a novel approach named FastDARTSDet, aiming to expedite the process of differentiable architecture joint search for both the backbone and FPN in the context of object detection. This innovative strategy tackled the architecture search problem through combinatorial optimization on graphs. It not only led to improved performance compared to manually designed networks but also reduced the computational requirements associated with such searches. FastDARTSDet showed competitive performance, particularly on the MS-COCO dataset, where it outperformed state-of-the-art Neural Architecture Search (NAS) methods for object detection. The efficiency of the proposed approach, which required only 4.2 GPU days, a substantial reduction compared to prior NAS methods designed for object detection, was emphasized. In addition, the evaluation mainly focused on the MS-COCO dataset. To further strengthen the results, a broader comparative analysis against other NAS methods and evaluations on diverse datasets would enhance the understanding of FastDARTSDet capabilities and their potential in various practical scenarios.

Zhou et al. [82] conducted a comprehensive survey that focused on efficient CNN and network compression methods applied to object detection. A detailed examination of the fundamental components, comprised of backbones, necks, and heads, was reported. Within this context, the survey emphasized the significance of efficient backbones, characterized by simplified structures and a reduced parameter count compared to traditional types, thereby making it suitable for resource-constrained devices. Various representatives of efficient object detectors were also introduced, showcasing methods applied to achieve efficiency. These approaches included strategies like increasing network depth and width or reducing parameter volume. However, the survey could benefit from a more in-depth analysis of the complex adjustments between efficiency and accuracy, specifically concerning different backbone architectures and compression methods.

This research attempted to address the challenges encountered in object detection using different approaches. However, the common goal was to create backbone architectures that excelled in computational, speed, and energy efficiencies, as well as preserve or enhance detection performance. These diverse approaches comprise a range of techniques, including one-shot aggregation, hierarchical connectivity formation, network compression, and the application of Neural Architecture Search (NAS) methods for automated architecture design.

The research also reported specific limitations, despite stating promising results in enhanced backbone efficiency. For instance, some failed to thoroughly examine the difference between accuracy and efficiency, especially when considering various resource constraints. Additionally, some others limited the model evaluations to specific object detection datasets and tasks. This limited scope raises questions about their findings' broader applicability and transferability to different datasets and diverse object detection tasks.

The progress achieved through these various approaches in enhancing backbone efficiency presents significant opportunities to optimize object detection performance across a range of applications. These advancements are particularly valuable for real-time object detection and for deploying models on resource-constrained devices. In this situation, the survey on efficient CNN and network compression methods plays a relevant role in providing comprehensive insights into the current state of progress and future research directions for the development of more efficient and practical backbone architectures designed for object detection. With these results and breakthroughs, the future of object detection holds great potential for practical implementations across diverse fields and applications.

B. RPN SIMPLIFICATION

In this section, RPN simplification was analyzed by investigating a collection of research studies that focus on optimizing RPN to enhance the efficiency of object detection. This research reported diverse strategies, including feature fusion, parameter compression, and innovative network architectures, all aimed at achieving a delicate balance between computational efficiency and accuracy in real-time object detection tasks. Through this comprehensive examination, the main objective is to uncover common themes and recent advancements and address existing challenges within the domain of simplified RPN. The main aim is to pave the way

for developing more efficient and practical solutions in object detection.

Kim et al. [83] proposed a novel neural network architecture for real-time object detection, which effectively balances high accuracy with computational efficiency. This approach was achieved through a thoughtful redesign of the feature extraction segment within the object detection pipeline. This redesign followed the lesser channels with more principle layers, meaning it used fewer channels per layer but stacked more to maintain accuracy. Major building blocks, e.g., concatenated ReLU, Inception, and HyperNet, were introduced to enhance the network performance further. In RPN, these blocks optimized efficiency by using only the first 128 channels in the feature map. The results of the experiments, conducted on well-established object detection benchmarks, showcased the effectiveness of PVANET. It achieved an impressive mAP of 83.8% and 82.5% on VOC2007 and VOC2012, respectively, securing the second-place ranking. Meanwhile, PVANET exhibited exceptional computational efficiency, with a requirement of only 750 ms and 46 ms per image on an Intel i7-6700K CPU with a single core and NVIDIA Titan X GPU, respectively. This was accomplished while consuming only 12.3% of the computational cost compared to ResNet-101, which secured victory in VOC2012. The research lacked a detailed analysis of the dissimilarity between accuracy and computational cost, and it failed to explicitly discuss potential limitations or the extent of its applicability to other datasets or real-world scenarios. PVANET represents a significant advancement in real-time object detection, effectively combining DNN with lightweight design principles, offering a promising avenue for future research in this field.

In 2018 Li et al. conducted research on efficient object detection for resource-limited embedded devices. The objective was to devise a method that could perform object detection effectively while accommodating the constraints of such devices [84]. This goal was achieved by introducing a two-stage object detection method, namely a subnetwork-efficient feature fusion module (EFFM) and a multi-scale dilation RPN to reduce the number of operations and parameters in a two-stage detector. The results showed that the proposed method outperformed Faster RCNN, based on VGG16, in terms of accuracy. This improved performance was achieved while using only half the computational operations and one-third of the parameters. The EFFM efficiently combines local channel information through the use of pointwise and grouped convolutions, while RPN improves recall rates by incorporating multi-scale dilation and global feature weighting techniques. Although the research mentioned the potential application of knowledge distillation to further reduce parameters, it emphasizes that this technique was limited to classification tasks. The effectiveness of the method was evaluated on PASCAL VOC datasets, showing promise for object detection on resource-limited embedded devices. Its performance on other datasets and tasks remains an area for further investigation.

The groundbreaking work by Chen et al. [85] led to the proposal of the Multi-strategy Region Proposal Network (MSRPN). This innovative architecture aims to enhance object detection performance by overcoming the limitations of the traditional RPN. To achieve this, MSRPN introduces four significant improvements. First, it incorporates a novel skip-layer connection network, which enhances pooling layers and combines multi-level features. This enhancement is aimed towards improving the quality of region proposals. Secondly, MSRPN introduced improved anchor boxes with adaptive aspect ratios and a well-distributed selection of scales. These enhancements were designed to reduce the number of predicted region proposals while increasing the efficiency of object localization, specifically for small object detection. Thirdly, MSRPN unifies the classification and regression layers into a single convolutional layer. This consolidation accelerates both training and testing speed while simultaneously reducing model complexity in the output layer. Lastly, the bounding box regression component of the multi-task loss function in RPN undergoes improvements in MSRPN. These enhancements are implemented to enhance the performance of bounding box regression. MSRPN distinguishes itself by outperforming five other object detection methods, achieving state-of-the-art mAP scores across PAS-CAL VOC 2007, 2012, and MS COCO datasets when coupled with the deep VGG-16 model. One outstanding feature is its superior performance in detecting small objects while maintaining a rapid processing speed of 6 frames per second compared to competitors. However, it is important to acknowledge several limitations within the paper's findings, namely, the evaluation primarily focuses on restricted datasets and relies exclusively on the deep VGG-16 model. It emphasizes the necessity for further validation across a wide range of datasets and models to ensure broader applicability of the method. MSRPN also requires significant computational resources, potentially limiting its applicability on lower-end devices. The research lacked a comprehensive analysis of the trade-offs between the proposed improvements, which could impact the interpretability of the results. Therefore, MSRPN presents a promising approach to object detection, achieving state-of-the-art performance while addressing the limitations of the traditional RPN. The four main improvements contributed to its success in object detection tasks. Further research is needed to validate the proposed method on a wider range of datasets and models while also addressing its limitations to ensure broader applicability.

Lin et al. introduced the Cropping Region Proposal Network (CRPN), a novel framework designed to improve object detection efficiency in large-scale remote sensing images [86]. It comprised a weak semantic RPN for rapidly identifying interesting regions and applied a dual-scale strategy to generate effective cropping regions. CRPN filtered invalid regions, reducing the computation burden and facilitating the precision of object detection. Moreover, it was modularized and easily integrated with mainstream

detectors, forming an end-to-end detecting framework. The effectiveness was proven on the public DOTA dataset, yielding superior object detection efficiency and accuracy results. There are certain limitations, namely the need for evaluation on diverse datasets, a detailed analysis of computational costs, comparisons with state-of-the-art methods, and further exploration of hyperparameter impacts. CRPN presents a promising approach for object detection in large-scale remote sensing images, potentially advancing the field of intelligent object detection systems.

Fan and Ling proposed a new approach for Real-Time Visual Tracking using Siamese Cascaded RPN [87]. This innovative framework was designed to address specific limitations observed in one-stage Siamese-RPN trackers, particularly the challenges in handling similar distractors and variations in object scale. To enhance its performance in complex backgrounds, CRPN incorporates hard negative sampling within a cascade architecture. The research also introduced a novel feature transfer block (FTB) module that improves the use of this feature across different layers, thereby enhancing the representation ability to discriminate between objects. Additionally, CRPN refines the target bounding box progressively through multiple regression stages, resulting in more precise localization. The efficacy of the framework is assessed based on six widely recognized benchmarks, where it showed state-of-the-art results and achieves real-time tracking performance. It is important to acknowledge certain limitations, including the evaluation conducted on a limited set of datasets and the absence of comparisons with other tracking methods. The general performance of CRPN shows its considerable potential in real-life visual tracking scenarios.

Unlike previous research, Shih at al. introduced a novel method in the book titled Real-Time Object Detection with Reduced RPN through Multi-Feature Concatenation [88]. The approach targeted real-time object detection and focused on mitigating memory and performance issues associated with neural networks while maintaining accuracy. It also adopted weight pruning techniques to compress network parameters and introduces optimizations to RPN. These optimizations included the use of 1×1 convolutions, slimmer channels, and dilated ones, all aimed at enhancing detection accuracy. To address any potential loss of accuracy due to pruning, a multi-feature concatenation technique combining several feature maps was incorporated, thereby ensuring the reduced RPN has sufficient information for precise region detection. The performance of the method was evaluated on both ZF-Net and VGG16, while the experimental results exhibited significant parameter compression (81.3% and 73% for ZF-Net and VGG16), and simultaneously improved detection accuracy (ranging from 2.2% to 60.2% and 2.6% to 69.1% for ZF-Net and VGG16, respectively). The innovative approach was efficiently and accurately used to conduct real-time object detection in computer vision applications.

In the following year, Siradjuddin et al. proposed a two-stage detection approach, known as Faster RCNN, for the detection of masked faces in images [89]. The method used RPN in its first stage to efficiently identify candidate regions. These were generated by sliding a small network over the convolutional feature map and combining anchor and bounding box offsets. In the second stage, these candidate regions are subjected to processing through an ROI Pooling layer for localization and classification using CNN architecture. However, the approach has certain limitations, namely the use of a restricted dataset, imbalanced data during the training process, and potential challenges in real-time applications. Further investigation and comparisons with other state-of-the-art methods for masked face detection are warranted. It was reported that faster RCNN showed promising results in the detection of masked faces, thereby paving the way for future research in this field.

Zhang et al. also proposed an inventive method to improve the quality of region proposals in weakly supervised object detection in 2021 [90]. The main contribution is the introduction of the Hierarchical Region Proposal Refinement Network (HRPRN), a system engineered to iteratively fine-tune region proposals through the use of multiple weakly supervised detectors. HRPRN comprised several critical components in its design, including image feature extraction, Region of Interest (RoI)-pooling layer, weakly supervised detector, hierarchical detector, and instance regression refinement models. This structured approach empowers HRPRN to gradually enhance the precision of object localization by iteratively perfecting region proposals. An important aspect of this proposal is the instance regression refinement model, which generates coordinate offsets sensitive to the objects in each stage, further improving the precision of localization. The evaluation of the method was conducted on the PASCAL VOC 2007 dataset, leading to a significant performance enhancement in terms of mAP and CorLoc when compared to the baseline method [1]. However, the research has some limitations inherent to its approach. One limitation lies in the fact that the evaluation is constrained to a relatively small dataset. This raises questions about the ability of the method to scale up and generalize effectively to larger datasets. The research also recognizes the computational cost associated with training multiple weakly supervised detectors in a staged manner. Irrespective of these limitations, HRPRN presented promising results. Further research is needed to address these limitations and explore the potential of the method when applied to larger datasets, along with comparisons to state-of-the-art methods.

Chen and Hao [91] conducted research to improve hardware-efficient object detection for embedded systems. The approach focused on achieving real-time inference with minimal energy consumption and limited hardware resources. The concept of a masked region proposal method was introduced to address the challenges associated with object detection in these constrained environments. This innovative approach generated rectangular regions of interest in regular shapes, effectively minimizing redundant computations during the detection process. Its effectiveness was

validated by applying the approach to various detection backbones, including SkyNet, ResNet-18, and UltraNet. Their evaluation comprised three single-object detection and tracking datasets (DAC-SDC, OTB100, and UAV123), showing the method's adaptability across diverse scenarios. Comparative analysis with existing region proposal methods showed that the masked approach offered simplicity and efficiency, requiring less feature extraction before the region proposal stage. An accelerator was designed to assess the method performance on the Xilinx ZCU106 FPGA, achieving a 1.3 × speedup and approximately 30% reduction in energy consumption with minimal accuracy loss. A design space exploration was conducted to show that the accelerator had the potential to achieve a theoretical speedup of 1.76× when used with masked region proposals. While the results of the proposed method showed promising potential for hardware-efficient object detection in embedded systems, it was also emphasized that the importance of further evaluation across a broad range of datasets and hardware types comprehensively assessed its generalizability and effectiveness.

Lastly, Zhu et al. introduced a simple and highly effective Siamese network called Siamese-ORPN for oriented visual tracking [92]. The main goal was to accurately estimate the position of a target object across subsequent video frames. To achieve this, Siamese-ORPN addressed the shortcomings of existing tracking methods by using oriented RPN and incorporating feature fusion. Oriented RPN played a critical role in generating high-quality-oriented proposals by predicting offsets and scales of related bounding boxes. Meanwhile, the feature fusion network leveraged different representations to predict a similarity map. The proposed method adopted an end-to-end training approach with a Siamese loss function. The results showed Siamese-ORPN's outstanding performance on the VOT2018 and VOT2019 challenges, achieving a commendable speed of 85 frames per second. This showcased the advantages of the method in terms of both accuracy and efficiency. It is essential to acknowledge that the evaluation was limited to VOT2018 and VOT2019 datasets. Further investigation is needed to explore Siamese-ORPN suitability for various tracking scenarios and datasets. Additionally, as a Siamese network-based approach, the proposed method demands a substantial amount of training data for optimal performance. Siamese-ORPN presents a promising avenue for oriented visual tracking, offering accurate and efficient results when benchmarked against these challenges.

The previous research represented various efforts in the field of object detection, sharing a common objective, namely improving computational efficiency and enabling real-time performance. This led to the introduction of novel neural network architectures, innovative feature extraction methods, and region proposal strategies, all with the aim of finding the right trade-off between accuracy and computational cost. A recurring theme in this research is the use of RPN to efficiently generate candidate regions for object detection. Several research used techniques such as feature fusion and concatenation to enhance the quality of representations, leading to improved detection results. These investigations had certain limitations, namely limited evaluations on specific datasets, lack of comparisons with state-of-the-art methods, and hardware-specific implementations. In-depth analyses regarding the trade-offs between accuracy and computational efficiency are often lacking. Collectively, these investigations offered promising directions for advancing real-time object detection and efficient model design in the future. It emphasized the importance of further validation on diverse datasets and hardware platforms to ensure the practical applicability of these advancements.

1) HEAD SIMPLIFICATION

In this section, the field of Head Simplification was explored, including a series of research on optimizing and streamlining object detection heads. These investigations present various innovative approaches, such as top-down refinement modules, auxiliary detection heads, and dynamic routing spaces, all united by the common goal of achieving a delicate balance between computational efficiency and accuracy in object detection tasks. Exploring these innovative strategies aims to uncover the significant advancements, challenges, and potential benefits associated with simplified object detection heads. Ultimately, the exploration aims to pave the way for developing more efficient and effective solutions in object detection.

Pinheiro et al. introduced an innovative architecture for object instance segmentation, known as SharpMask [93]. This approach enhanced feedforward networks by incorporating top-down refinement modules, which enabled the generation of highly accurate object masks by using features from all network layers. The SharpMask head architecture operates in two stages. First, it generates a coarse mask encoding during a feedforward pass, which is then refined in a top-down pass by leveraging features from progressively lower layers. An extensive investigation of the various factors influencing the network accuracy as conducted, including input size, pooling layers, stride density, model depth, and feature channels. SharpMask achieved state-of-the-art performance in object proposal generation, showing improvements of 10 to 20% in average recall compared to other setups while running 50% faster than the original DeepMask network. Despite this promising performance, the research emphasized on several areas that required further attention, such as evaluating the model on additional datasets, analyzing its computational complexity, addressing robustness to lighting changes and occlusions, as well as handling scenarios with limited training data. The proposed refinement approach in SharpMask showed significant versatility and adaptability to other pixel-labeling tasks, emphasizing its potential to advance object instance segmentation in terms of both accuracy and speed.

Jin et al. proposed a concept known as the auxiliary detection head (ADH), designed to enhance the performance

of one-stage object detectors after four years [94]. This ADH introduces implicit two-stage cascaded regression in a single detection head, including classification and regression subnets. The main goal is to refine object localization by adjusting output boxes to support ground truth, thereby facilitating learning more robust features. One significant advantage of this approach is its seamless integration into state-of-the-art object detection frameworks alongside the existing prediction branch, allowing for joint training with the original detection head. During inference, the ADH can be removed without affecting the main detector head, leading to reduced model size and faster inference time. The evaluation conducted on the Pascal VOC and COCO datasets consistently shows performance improvement over the baseline without introducing additional parameters at inference time. However, the research acknowledges certain limitations, such as the evaluation being limited to two datasets, the absence of comparisons with other two-stage cascaded regression methods, and the lack of detailed computational cost analysis and interpretability impact. Further research is essential to assess the effectiveness of the method on diverse datasets and compare it with other state-of-the-art object detection methods.

In the same year as the work of Jin Guozheng, Song et al. introduced the Fine-Grained Dynamic Head, offering a new approach to fine-grained object representation in object detection [95]. It has the unique capability to select a pixel-level combination of features from various scales within the Feature Pyramid Network (FPN) for each instance. This optimizes the use of multi-scale features while simultaneously reducing computational costs. To achieve this, the proposed method replaces the conventional head for FPN with a fine-grained dynamic routing space. This routing space dynamically allocates pixel-level sub-regions from multiple FPN stages. The method also incorporated a spatial gate featuring a novel activation function, further enhancing its computational efficiency. Extensive experiments on state-of-the-art detection benchmarks consistently exhibit the effectiveness and efficiency of the Fine-Grained Dynamic Head. It outperforms the conventional head, achieving state-of-the-art results with reduced computational overhead. However, to provide a more comprehensive understanding, the research could benefit from clearer insights into its performance on datasets outside those evaluated and potential computational resource requirements compared to the conventional head. The Fine-Grained Dynamic Head presents promising new avenues for object detection, hinting at future advancements in the field.

Miao et al. proposed a novel solution known as the Generalized Representation Reconstruction Head (RRHead) for object detection frameworks [96]. The main aim was to independently optimize both fully connected and convolutional-based detection heads, with a dual focus on enhancing the representation of feature-label mappings, including improving the encoding of location information.

An outstanding feature of RRHead is its effortless integration into existing detection frameworks, eliminating the need for any additional modifications. This integration not only streamlines the implementation process but also enhances the rationality and reliability of the detection head representation. The RRHead comprises three main components: Multi-Scale Feature Representation (MSFR), Location Sensitivity Enhancement Representation (LSER), and Feature Location Consistency Mapping (FLCM). MSFR uses a pyramid pooling module to extract features at different scales, thereby enhancing the representation of multi-scale features. This feature is essential for capturing information across different levels of granularity. LSER is dedicated to optimizing the encoding of location information through the use of a sensitivity module. This component is critical in preserving spatial details relevant to precise object localization in detection tasks. FLCM takes the original feature map and enhances it to use the advantages of multi-scale features and location data from both fully connected and convolutional-based detection heads. This refinement significantly improves the capability to map features to labels effectively. Depending on the specific type of detection head embedded, RRHead exhibits the ability to retain location sensitivity representation information while simultaneously enhancing feature-label space mapping in fully connected and convolutional-based detection heads. The effectiveness of RRHead is shown through extensive experiments conducted on challenging benchmarks, establishing it as a powerful tool for improving the detection performance of existing frameworks and achieving new state-of-the-art results. Although the research does not explicitly address the limitations of this method, potential concerns include its applicability to all object detection frameworks and variations in computational costs depending on the specific architecture and dataset used. RRHead offers valuable insights into enhancing the design of object detection heads and represents a promising advancement in the field.

The research by Zhu et al. [97] aimed to enhance object detection performance in drone-captured scenarios. This led to the introduction of TPH-YOLOv5, an improved version of YOLOv5, with significant innovation, including the integration of Transformer Prediction Heads (TPH) into the YOLOv5 framework. This integration is relevant for achieving precise object localization, particularly in scenes with high object density. TPH uses self-attention mechanisms to explore the model prediction potential, while CBAM (Convolutional Block Attention Module) enriches its ability to identify regions of interest within large image coverage. The model classification accuracy, especially for visually similar categories, was enhanced through the inclusion of a self-trained classifier. To further enhance its performance, a combination of data augmentation techniques, including MixUp, Mosaic, and traditional methods, was adopted. A set of strategic filtering techniques specifically designed for object detection in drone-captured scenarios was also used.

TPH-YOLOv5 exhibited state-of-the-art performance on the VisDrone2021 test-challenge dataset, surpassing the previous state-of-the-art model (DPNetV3) by 1.81% and competing with the top-ranking model in the VisDrone2021 DET challenge. However, the research acknowledged potential limitations, such as its performance on non-drone images, computational resource requirements, training time, and interpretability. As the research strived to enhance object detection in drone scenarios, it became crucial to explore its adaptability to other image types and consider real-time application requirements.

Dai et al. (2021), introduced a novel dynamic head framework aimed at improving object detection performance through the integration of attention mechanism [98]. It comprised three attention mechanisms, namely scale, spatial, and task-aware, which collectively capture diverse information within the feature tensor. This enhanced the representation ability of object detection without introducing computational overhead. The procedure was effortlessly integrated into existing detectors by applying scale and spatial-aware attention to the feature pyramid, while task-aware replaced fully connected layers. The results of the evaluation conducted on the COCO benchmark solidified the dynamic head as the new state-of-the-art, showcasing its superior performance. Though the investigation does not explicitly mention limitations, further exploring its performance across different datasets and architectures is necessary. It is intended to provide valuable insights into designing attentions for improved performance and introduces a new perspective of head design.

Xia et al. proposed CBASH, a new method for weakly supervised object detection, with a focus on the Advanced Selection Heads (ASH) as the central component [99]. ASH was devised to improve the quality and quantity of positive object proposals by using a two-stage method. First, a coarse selection head captures the most informative features from the backbone network, generating initial proposals. Furthermore, a fine selection head further refines these proposals by picking the most informative ones. CBASH effortlessly incorporates ASH into the entire model, facilitating joint training with a standard binary cross-entropy function. Ablation research validated its effectiveness, exhibiting significant performance improvements, particularly in terms of mAP@50. ASH shows potential for addressing the challenges of weakly supervised object detection. Further research is needed to evaluate its limitations and compare the performance with other state-of-the-art methods across a wider range of scenarios.

Yi Shi et al. introduced an innovative approach, namely MHD-Net, a novel lightweight traffic object detection network [100]. The main focus of this research is to enhance the performance of traffic object detection by refining its head method and configuration. MHD-Net introduces a matching strategy between the detection head and object distribution. This strategy guides the rational configuration of the detection head to effectively detect objects at various scales. The research advocates a cross-scale configuration guideline, suggesting the replacement of multiple detection heads with only two. The approach balances model parameters, FLOPs (floating-point operations per second), detection accuracy, and speed, thereby enhancing the model's efficiency. A receptive field enlargement method, combining dilated convolution modules with shallow and deep supervision, was also introduced. This combination led to improved detection accuracy, specifically for small objects. As a result, the proposed MHD-Net achieves state-of-the-art performance on benchmark datasets such as BDD100K and ETFOD-v2 for traffic object detection. Further investigation is required to determine its generalization to other datasets and object detection tasks, specifically those including complex-shaped objects or occlusions. Its suitability for real-time applications with high frame rates needs to be carefully considered.

Jiang and Gu proposed a new Gating Head (G-Head) in the context of one-stage object detection. The primary goal was to enhance the interaction between different tasks and promote effective multi-task learning [101]. The G-Head was specifically designed to address the limitations associated with the conventional parallel head structures. It consists of three relevant modules Multi-Scale Aggregation (MSA), Multi-Aspect Learning (MAL), and Gating Selector (GS). The MSA module plays a significant role in acquiring shared information by aggregating features from multiple scales. Meanwhile, the MAL module focused on aspect-specific features using its convolutional filters. The function of the GS module is to adaptively select the most informative features from different aspects and scales. By breaking down the multi-task learning problem into distinct aspects, the G-Head simplifies the training process and significantly achieves significant performance improvements. It also achieves these gains while requiring fewer parameters and performing fewer floating-point operations per second (FLOPs). Through extensive experiments conducted on the challenging MS COCO dataset, the proposed G-Head sets a new state-of-the-art benchmark with an impressive 48.7 Average Precision (AP) score under single-model and single-scale testing conditions. Although the research does not explicitly list its limitations, future investigation is recommended to analyze the method's performance on alternative datasets and compare its computational efficiency with other state-of-the-art techniques. As a result, the G-Head method is an effective and efficient approach for one-stage object detection. It shows great promise for advancing multi-task learning in the context of object detection, signifying a significant advancement in this field.

Based on the research conducted by Zhu et al., a DualDA-Net framework was proposed for the purpose of cross-domain object detection in remote sensing imagery [102]. This framework is specifically designed to address the challenges posed by the biased information that commonly occurs between the source and target domains. To accomplish this,

DualDA-Net adopted a teacher-student framework with dual detection heads. These heads serve the dual purpose of generating pseudo-labels for the target domain data and progressively refining them. The framework consists of two main components, namely coarse-to-fine consistency alignment (CCA) and dual-head co-training (DHCT). CCA focuses on supporting the feature distribution between the source and target domains, effectively mitigating the shift issue. On the other hand, DHCT incorporated dominant and affiliated heads to reduce biased information and improve the quality of pseudo-labels. This framework find it difficult to accurately detect certain object categories with unique sizes and aspect ratios. Addressing variations in category distribution between domains is also an important consideration. The presence of noisy labels in predictions can impact performance, besides future investigation in this area could explore oriented object detection techniques and domain-specific strategies to enhance results across all categories. The DualDA-Net framework shows promise in cross-domain object detection in remote sensing imagery, achieving success in the target domain.

Recent research in the field of object detection head methods share a common objective, to enhance object detection performance through innovative architectural designs and attention mechanisms. These approaches aim to strike a delicate balance, achieving both high accuracy and computational efficiency while simultaneously tackling a range of challenges such as domain shift, weakly supervised object detection, and scenario-specific complexities. The proposed methods offer a variety of solutions, including the integration of top-down refinement modules, auxiliary detection heads, dynamic routing spaces, fine-grained dynamic routing, etc. Collectively, these innovations work to improve the overall effectiveness of object detection systems. Additionally, significant progress has been made in refining attention mechanisms in this research. These mechanisms, including scale, spatial, and task-aware attention, have exhibited promising results in augmenting the representation capabilities of object detection models. This research collectively contributes valuable insights and techniques for advancing the field of object detection, paving the way for more efficient and accurate models in the future.

VI. DISCUSSION

Object detection has seen remarkable improvements in speed and accuracy, mainly due to developments in deep learning and the availability of large datasets. Yet, the substantial computational and memory demands of deep learning methods pose challenges for deploying object detection on edge devices with limited hardware resources.

In this work, we mainly focus on the state of the art of model simplification techniques to enable object detection on resource-constrained edge devices. Table 1 on page 5 presents the comparable survey paper related to our work. This survey closes the existing gap among previous reviews. Object detection development during the past two decades was provided by [13]. However, they did not put an emphasis on the simplification techniques and edge implementation. The edge of object detection implementation is partially discussed in [39] and [42]. While [41] explored the available open source edge device related project and their energy efficiency. Moreover, [40] discussed the general approach of edge computing including the coordination of edge devices and cloud computing, but they do not comprehensively cover the model compression techniques in their review.

Edge devices are designed to operate efficiently under limited conditions, focusing on features like compact size, lightweight, and low power consumption. This focus often leads to trade-offs, particularly regarding limited memory and processing power (including CPUs and GPUs). These limitations can be challenging for implementing near-edge computing. To address these challenges, previous research has focused on adapting and simplifying model complexities. An overview of hardware configurations for various edge devices is provided in Table 2. Devices such as the Raspberry Pi primarily use CPUs, while others like the Jetson Nano and Xavier combine CPUs with GPUs, offering memory capacities of 4 GB and 8 GB, respectively. Additionally, there are devices like the FPGA Virtex 7 VC707, which have minimal memory, starting at just 18 kB. These hardware specifications greatly impact the computational abilities and power consumption of the devices.

When deploying deep learning object detection models on edge devices, models like YOLOv4 with the CSPDarknet53 backbone face significant challenges due to their extensive model sizes and the consequent demands on memory and computing resources. For example, processing a 416×416 image with YOLOv4 involves managing over 60 million parameters during inference. These models require substantial memory not only for storing the model itself but also for holding the results computed during inference. A comparative analysis of various object detection backbones, which vary in size from 26 MB to 12 GB, is detailed in Table 3. This variation is influenced by factors like kernel size, number of layers, and input image dimensions.

The primary function of the backbone is to transform the input into a feature space, and it typically accounts for the largest portion of the model size compared to other components, such as the head network. The size of a model in neural network-based methods has two significant consequences, namely, memory footprint and processing unit requirements. Models essentially consist of learned weight parameters acquired during training. These weight parameters are essential for making inferences to predict outputs. Before making inferences, the model must be loaded into memory, which requires more memory for larger ones. Inferences involve forward propagation, where the input signal is multiplied by the weight of the model to predict either numerical values (regression) or the class (classification).

Edge devices usually feature processing units, such as CPUs and GPUs, tasked with these computational processes.

TABLE 2. Various edge devices capacity.

Edge Devices	CPUs	GPUs	Memory
Raspberry PI 3B [61], [66]	ARM Cortex A53, A57- Quad Core 1.2GHz	-	1GB
Raspberry PI 4B [72]	Quad-core Cortex-A72 (ARMv8) 64-bit SoC @ 1.8GHz	-	4GB
Nvidia Jetson Nano [65], [71]	Quad-core ARMCortex-A57 @1.43GHz	128Core Maxwell 472 GFlops	4GB
Nvidia Jetson TX-2 [61]	64-Bit Denver 2 with 6 CoreCortex- A57 @1.43GHz	256Core Pascall 1.3TFLOPS	8GB
Nvidia Jetson Xavier-Nx [71]	6-core NVIDIA Carmel ARM® v8.2 64-bit	384-Core NVIDIA Volta21 TOPS	8GB
FPGA Virtex- 7 VC707 [70]	0.2GHz	-	18Kb
FPGA Adm 7v-3 [22]	0.2GHz	-	6.6MB BRAM
FPGA ZCU106 [91]	Logic cells 504K	-	6.2MB
Google Pixel 2 [20]	Qualcomm Snapdragon 835 LITTLEcore 8-Core	Qualcomm® Adreno™ 540	4GB
Google Pixel 3 [20]	Qualcomm Snapdragon 845 BIGcore 8-Core	Qualcomm® Adreno™ 630	4GB

TABLE 3. Object detection backbone.

Object Detection Backbone	Size
DenseNet-169 [61], [96]	400MB
DenseNet-201 [61]	Based on GPU Memory [99]
VGG16 [83]-[85], [88]-[90], [96], [99]	528 MB (depends on various factors)
YOLO [70], [71]	not explicitly mentioned
YOLOv2 [65], [70]	not explicitly mentioned
YOLOv3 [65]	not explicitly mentioned
YOLOv4 [72]	
YOLOv5 [97], [100]	
ResNet-20 [66]	not explicitly mentioned
ResNet-18 [21], [91]	45 MB
ResNet-50 [62], [63], [21], [22], [83], [92], [95], [98]	219 MB
ResNet-101 [62], [63], [76], [83], [94], [95]	171 MB
ResNet-110 [66]	not explicitly mentioned
ResNet-152 [61]	not explicitly mentioned
ResNeXt-101-DCN [98]	not explicitly mentioned
MobileNet [20]	increases as the network becomes deeper [107]
MobileNet-v2 [61], [63]	26MB
EfficientNet-B0 [63]	4 GB
EfficientNet-B1 [95]	not explicitly mentioned
GoogLeNet [84]	not explicitly mentioned
ZF-Net [88]	96 MB per image
SkyNet [91]	6 GB
RefineDet [94]	not explicitly mentioned
Inception-v3 [61]	less than 12 GB memory
MnasNet [61]	not explicitly mentioned

Considering the substantial size of backbones, efforts have been made towards their replacement or reduction. For instance, in YOLOv4 and YOLOv4 Tiny, the original 53-layer backbone is replaced with a more compact 9-layer version. Modifications to the head network are also explored, though these tend to be less extensive due to the head network generally having fewer layers than the backbone.

There are three well-known deep learning compression techniques, namely knowledge distillation, pruning, and quantization, as shown in Figure 3 and elaborated in section IV. CNN is a fundamental component in the field of deep learning object detection methods, including well-known methods such as RCNN, YOLO, and SSD. Most of the detection methods implement transfer learning of common CNNs such as VGG 16 in the RCNN family, YOLO VGG16, and SSD. These CNN models play a critical role in the process, mainly by transforming the input image into a smaller feature space. This transformation makes subsequent tasks computationally lighter while still retaining important information from the input image to maintain high detection accuracy. The primary differences among object detection methods such as RCNN, YOLO, and SSD lie in when the feature extraction would be executed. For instance, in the Early RCNN family, feature extraction occurs after computing region proposals, while in YOLO it is initiated at the initial stage of object detection.

YOLO and SSD follow a one-stage approach where, after feature extraction, the subsequent tasks involve regression of object properties, such as location and size, along with object classification. Regression and classification tasks can be accomplished using various methods, including SVM and neural networks. More recently, in YOLO, bounding box prediction and object classification are both carried out by neural networks.

These object detection methods take feature space as input and produce numerical bounding box predictions and object class labels as output. To restructure the complexity of object detection methods, simplifications can be made in the network head, typically constituting the backbone. Replacing the backbone with a smaller CNN size often comes at the cost of feature quality, which can reduce detection accuracy. A significant challenge is to maintain detection accuracy while reducing network size, especially for deployment on edge devices. Various compression techniques such as quantization, pruning, and knowledge distillation have been explored to minimize CNN while ensuring high accuracy compared to the original algorithm before compression. Knowledge distillation has shown promising results as reported in prior research [61], [62], and [63], while pruning and quantization have also been investigated by [66] and [72].

MSCOCO and PASCAL VOC serve as widely recognized benchmark datasets, extensively used in various research to evaluate system performance in the field of object detection. Prior research explored different variations of techniques such as knowledge distillation, network pruning, and quantization to enhance the efficiency of the proposed models when working with these datasets. The primary evaluation metric commonly used in this context is mAP. In the past, there has been a significant surge in research attention directed towards compressing object detection models. However, certain challenges within this context remain partially unresolved. Specifically, the pursuit of improving mAP scores and reducing computation time remains highly prioritized. This is critical for achieving real-time performance on devices with limited computational resources.

TABLE 4. State-of-the-art research papers on object detection compression method.

Authors	Years	Dataset	Approach	Results
Matsubara et al. [61]	2020	ImageNet (ILSVRC)	Head Network distillation	Val accuracy increasead. DenseNet-169=72.03(+0.159) DenseNet-201=73.62(+1.750), ResNet-152=75.13(+3.259) InceptionV3=75.78(+3.910) and training speed also increased.
Liu et al. [62]	2021	COCO, PASCAL VOC	Mutual Information Knowledge Distillation	COCO=68.4mAP, VOC=77.9mAP -
Kang et al. [63]	2021	MS-COCO	Instance-Conditional Knowledge Distillation (ICD)	RetinaNet with a ResNet- 50 backbone improved from 37.4 to 40.7 mAP (+3.3), ICD Improves up to 4 AP for instance segmentation, Improved in efficient Backbone with AP ranging from 2.6 to 5.2.
Chawla et al. [26]	2021	DIODE synthesizes images	Deep Inversion for Object Detection (DIODE)	Data-free distillation for object detection using these synthesized images yields a significant improvement (0.450 mAP) compared to out-of-domain proxy datasets (0.313 mAP) and are competitive with same domain proxy datasets (0.466 mAP).
Cheng et al. [66]	2022	CIFAR10, CIFAR100, ImageNet	novel neuron manifold distillation (NMD)	71.92 accuracy on top-1 re- sult and 74.58 accuracy on top-1 with self-knowledge distillation, outperforming the baseline model
Liang et al. [71]	2022	COCO2017, KITTI	SPPFPN modules (SPPNet and FPN), pruning feature extraction network and compression feature fusion network	26.6 FPS, mAP 47.3, and sizes 25,27MB on the COCO2017, mAP 0.82, 40.6FPS, and 24.48MB on the KITTI Dataset.
Liberatori et al. [72]	2022	Mask-Detection-Dataset	YOLOv4-tiny with ResBlock-D modules, Block Attention Module, Pruning and Dynamic Quantization	mAP of the dynamic quantized is 0.574 (drop only 0.044 from baseline).
Xia et al. [99]	2022	PASCAL VOC 2007 and 2012	propose an efficient network head called Advanced Selection Heads to generate more pseudo ground truth boxes with high confidence for each category, which can make more relevant proposals be trained.	This model gets the best performance in the class of airplanes (65.5 mAP), bikes (74.4 mAP), boats (36.3 mAP), chairs (26.9 mAP), motor (70.4 mAP), and sheep (58.9 mAP). 9 categories for AP50.
Y Zhu et al. [102]	2023	NWPU VHR-10, DOTA WHU Buildings Change Detection, RarePlanes, and FAIR1M	DualDA-Net uses CCA to extract features and uses DHCT with dual heads for pseudo-label rectification. By aligning the distribution of the source and target domains.	The DualDA-Net got 37.83 mAP achieving the best performance in both the experiments with Res101 as the backbone and 39.50 with Res101+FPN.

It is essential to acknowledge that there is no universal solution that works effortlessly in every situation, despite the progress made by some research in simplifying object detection methods. Evaluating the effectiveness of compression approaches relies on two critical considerations, improving latency during inference on target devices and managing potential drops in mAP. It is relevant to emphasize the importance of evaluations carried out in specific use-case scenarios. The widely used standard datasets such as COCO or PASCAL VOC have certain advantages; however, they do not completely capture the complexities of real-world applications, for example, the case of an agricultural drone designed for automated fruit harvesting. In this scenario, the diversity of object types is significantly smaller compared to COCO or PASCAL VOC. Therefore, assessing the performance of an object detector using general datasets alone may not suffice in providing insights into its real-world effectiveness.

VII. CONCLUSION

This research presented a comprehensive review of deep learning-based object detection methods. The study emphasizes that CNN architectures form the backbone of these models, and the overall size of an object detection model is largely influenced by the chosen backbone. Compressing the backbone can significantly reduce the model's size. Beyond the backbone, the compression of the head in object detection methods has also been a focus in recent advancements.

The main objective was to investigate the state of the art in object detection compression and to identify areas for further research. This exploration included three key deep-learning compression techniques: knowledge distillation, pruning, and quantization. Regarding knowledge distillation, future studies are encouraged to experiment with different teacher-student model combinations. With pruning, the challenge lies in determining the most effective strategy for deep neural networks. Previous studies have also examined quantization techniques and their impact on the performance of object detection models.

Moreover, the varied responses of different target devices indicate a need for more research to understand these discrepancies fully. The specific use case is another crucial factor; while some algorithms may perform well on general

detection datasets like MS COCO, their effectiveness might vary in scenarios with a different number or type of target objects. Understanding these nuances is vital for developing tailored and efficient object detection solutions.

ACKNOWLEDGMENT

The authors are grateful to the Research Directorate of Universitas Amikom Yogyakarta for the support provided to this project.

REFERENCES

[1] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[2] J. Wang, J. Ding, H. Guo, W. Cheng, T. Pan, and W. Yang, "Mask OBB: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images," *Remote Sens.*, vol. 11, no. 24, p. 2930, Dec. 2019, doi: 10.3390/rs11242930.

[3] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 3–22, Nov. 2018, doi: 10.1016/j.isprsjprs.2018.04.003.

[4] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 39–51, Jan. 1998.

[5] H. Wang, C. Pan, X. Guo, C. Ji, and K. Deng, "From object detection to text detection and recognition: A brief evolution history of optical character recognition," *WIREs Comput. Statist.*, vol. 13, no. 5, p. e1547, Jan. 2021, doi: 10.1002/wics.1547.

[6] F. Gao, C. Wang, and C. Li, "A combined object detection method with application to pedestrian detection," *IEEE Access*, vol. 8, pp. 194457–194465, 2020, doi: 10.1109/ACCESS.2020.3031005.

[7] S. C. H. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, and Q. Wu, "LOGO-Net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks," 2015, *arXiv:1511.02462*.

[8] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, "A review of video object detection: Datasets, metrics and methods," *Appl. Sci.*, vol. 10, no. 21, p. 7834, Nov. 2020.

[9] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7210–7218.

[10] A. Ramachandran and A. K. Sangaiah, "A review on object detection in unmanned aerial vehicle surveillance," *Int. J. Cognit. Comput. Eng.*, vol. 2, pp. 215–228, Jun. 2021.

[11] Z. Li, M. Dong, S. Wen, X. Hu, P. Zhou, and Z. Zeng, "CLU-CNNs: Object detection for medical images," *Neurocomputing*, vol. 350, pp. 53–59, Jul. 2019.

[12] A. Kaur, Y. Singh, N. Neeru, L. Kaur, and A. Singh, "A survey on deep learning approaches to medical images and a systematic look up into real-time object detection," *Arch. Comput. Methods Eng.*, vol. 29, no. 4, pp. 2071–2111, Jun. 2022.

[13] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 9905, Oct. 2016, pp. 21–37.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[17] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks," *J. Mach. Learn. Res.*, vol. 22, pp. 1–124, Jan. 2021.

[18] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding," in *Proc. ICLR*, 2016, pp. 1–14.

[19] S. Ghosh, S. K. K. Srinivasa, P. Amon, A. Hutter, and A. Kaup, "Deep network pruning for object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3915–3919.

[20] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[21] R. Li, Y. Wang, F. Liang, H. Qin, J. Yan, and R. Fan, "Fully quantized network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2805–2814.

[22] C. Ding, S. Wang, N. Liu, K. Xu, Y. Wang, and Y. Liang, "REQ-YOLO: A resource-aware, efficient quantization framework for object detection on FPGAs," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, Feb. 2019, pp. 33–42.

[23] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-S. Hua, "Quantization networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7300–7308.

[24] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 742–751.

[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[26] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez, "Data-free knowledge distillation for object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3288–3297.

[27] X. Liu and Z. Zhu, "Knowledge distillation for object detection based on mutual information," in *Proc. 4th Int. Conf. Intell. Auto. Syst. (ICoIAS)*, May 2021, pp. 18–23, doi: 10.1109/ICOIAS53694.2021.00011.

[28] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video," 2017, *arXiv:1709.05943*.

[29] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 31, Dec. 2018, pp. 1967–1976. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3326943.3327124

[30] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[32] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*.

[33] P. Subedi, J. Hao, I. K. Kim, and L. Ramaswamy, "AI multi-tenancy on edge: Concurrent deep learning model executions and dynamic model placements on edge devices," in *Proc. IEEE 14th Int. Conf. Cloud Comput. (CLOUD)*, Chicago, IL, USA, Sep. 2021, pp. 31–42, doi: 10.1109/CLOUD53861.2021.00016.

[34] J. Hao, P. Subedi, L. Ramaswamy, and I. K. Kim, "Reaching for the sky: Maximizing deep learning inference throughput on edge devices with AI multi-tenancy," *ACM Trans. Internet Technol.*, vol. 23, no. 1, pp. 1–33, Feb. 2023.

[35] K. R. Rajput, C. D. Kulkarni, B. Cho, W. Wang, and I. K. Kim, "EdgeFaaSBench: Benchmarking edge devices using serverless computing," in *Proc. IEEE Int. Conf. Edge Comput. Commun. (EDGE)*, Jul. 2022, pp. 93–103.

[36] T. Belabed, M. G. F. Coutinho, M. A. C. Fernandes, C. V. Sakuyama, and C. Souani, "User driven FPGA-based design automated framework of deep neural networks for low-power low-cost edge computing," *IEEE Access*, vol. 9, pp. 89162–89180, 2021, doi: 10.1109/ACCESS.2021.3090196.

[37] S. Rivas-Gomez, A. J. Pena, D. Moloney, E. Laure, and S. Markidis, "Exploring the vision processing unit as co-processor for inference," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2018, pp. 589–598.

[38] M. Lang, "EGX platform for edge computing," NVDIA, Singapore, 2019. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/en_sg/ai-innovation-day-2019/assets/pdf/7_EGX-Platform-for-Edge-Computing.pdf

[39] Z. Huang, S. Yang, M. Zhou, Z. Gong, A. Abusorrah, C. Lin, and Z. Huang, "Making accurate object detection at the edge: Review and new approach," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2245–2274, Sep. 2021, doi: 10.1007/s10462-021-10059-3.

[40] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020, doi: 10.1109/COMST.2020.2970550.

[41] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proc. IEEE*, vol. 107, no. 8, pp. 1537–1562, Aug. 2019, doi: 10.1109/JPROC.2019.2920341.
[42] P. Kang and A. Somtham, "An evaluation of modern accelerator-based edge devices for object detection applications," *Mathematics*, vol. 10, no. 22, p. 4299, Nov. 2022, doi: 10.3390/math10224299.
[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009, doi: 10.1007/s11263-009-0275-4.
[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 8693, 2014, pp. 740–755.
[45] A. Schmidt and A. Kasiski, "The performance of the Haar cascade classifiers applied to the face and eyes detection," in *Computer Recognition Systems 2* (Advances in Soft Computing). Berlin, Germany: Springer, 2007, pp. 816–823.
[46] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 606–613.
[47] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[48] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[49] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Aug. 2010.
[50] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2241–2248.
[51] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured HOG-LBP forp object localization," in *Proc. CVPR*, Colorado Springs, CO, USA, Aug. 2011, pp. 1393–1400, doi: 10.1109/CVPR.2011.5995678.
[52] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013, doi: 10.1007/s11263-013-0620-5.
[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012.
[54] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[55] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.
[57] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*.
[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[59] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Mali, Jun. 2014, pp. 580–587.
[60] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[61] Y. Matsubara, D. Callegaro, S. Baidya, M. Levorato, and S. Singh, "Head network distillation: Splitting distilled deep neural networks for resource-constrained edge computing systems," *IEEE Access*, vol. 8, pp. 212177–212193, 2020.
[62] G. Li, X. Li, Y. Wang, S. Zhang, Y. Wu, and D. Liang, "Knowledge distillation for object detection via rank mimicking and prediction- guided feature imitation," in *Proc. Conf. Artif. Intell.*, 2022, vol. 36, no. 2, pp. 1306–1313.
[63] Z. Kang, P. Zhang, X. Zhang, J. Sun, and N. Zheng, "Instance-conditional knowledge distillation for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 16468–16480.
[64] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez, "Data-free knowledge distillation for object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3289–3298.
[65] M. Bharadhwaj, G. Ramadurai, and B. Ravindran, "Detecting vehicles on the edge: Knowledge distillation to improve performance in heterogeneous road traffic," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3191–3197.
[66] Z. Tao, Q. Xia, S. Chen, and Q. Li, "An efficient and robust cloud-based deep learning with knowledge distillation," *IEEE Trans. Cloud Comput.*, vol. 11, no. 2, pp. 1733–1745, Mar. 2022.
[67] D. Li, T. Salonidis, N. V. Desai, and M. C. Chuah, "DeepCham: Collaborative edge-mediated adaptive deep learning for mobile object recognition," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 64–76.
[68] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.
[69] R. Hinami and S. Satoh, "Large-scale R-CNN with classifier adaptive quantization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 403–419.
[70] D. T. Nguyen, T. N. Nguyen, H. Kim, and H.-J. Lee, "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1861–1873, Aug. 2019.
[71] S. Liang, H. Wu, L. Zhen, Q. Hua, S. Garg, G. Kaddoum, M. M. Hassan, and K. Yu, "Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25345–25360, Dec. 2022.
[72] B. Liberatori, C. A. Mami, G. Santacatterina, M. Zullich, and F. A. Pellegrino, "YOLO-based face mask detection on low-end devices using pruning and quantization," in *Proc. 45th Jubilee Int. Conv. Inf., Commun. Electron. Technol. (MIPRO)*, May 2022, pp. 900–905.
[73] R. Li, Y. Wang, F. Liang, H. Qin, J. Yan, and R. Fan, "Fully quantized network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2810–2819.
[74] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*. Boca Raton, FL, USA: CRC Press, 2022, pp. 291–326.
[75] Y. Lee, J.-W. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and GPU-computation efficient backbone network for real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 752–760.
[76] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, "DetNAS: Backbone search for object detection," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 32, Dec. 2019, pp. 6642–6652, Art. no. 596. [Online]. Available: https://dl.acm.org/doi/10.5555/3454287.3454883
[77] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.
[78] C. Chen, M. Liu, X. Meng, W. Xiao, and Q. Ju, "RefineDetLite: A lightweight one-stage object detection framework for CPU-only devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 2997–3007.
[79] W. Hong and A. Song, "Improving deep object detection backbone with feature layers," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2021, pp. 91–105.
[80] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 280–296.
[81] C. Wang, X. Wang, Y. Wang, S. Hu, H. Chen, X. Gu, J. Yan, and T. He, "FastDARTSDet: Fast differentiable architecture joint search on backbone and FPN for object detection," *Appl. Sci.*, vol. 12, no. 20, p. 10530, Oct. 2022.
[82] Y. Zhou, L. Xia, J. Zhao, R. Yao, and B. Liu, "Efficient convolutional neural networks and network compression methods for object detection: A survey," *Multimedia Tools Appl.*, vol. 2023, pp. 1–43, Jun. 2023.
[83] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, "PVANET: Deep but lightweight neural networks for real-time object detection," 2016, *arXiv:1608.08021*.
[84] J. Li, K. Peng, and C.-C. Chang, "An efficient object detection algorithm based on compressed networks," *Symmetry*, vol. 10, no. 7, p. 235, Jun. 2018.

[85] Y.-P. Chen, Y. Li, G. Wang, and Q. Xu, "A multi-strategy region proposal network," *Expert Syst. Appl.*, vol. 113, pp. 1–17, Dec. 2018.
[86] Q. Lin, J. Zhao, Q. Tong, G. Zhang, Z. Yuan, and G. Fu, "Cropping region proposal network based framework for efficient object detection on large scale remote sensing images," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1534–1539.
[87] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7944–7953.
[88] K.-H. Shih, C.-T. Chiu, J.-A. Lin, and Y.-Y. Bu, "Real-time object detection with reduced region proposal network via multi-feature concatenation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2164–2173, Jun. 2020.
[89] I. A. Siradjuddin and A. Muntasa, "Faster region-based convolutional neural network for mask face detection," in *Proc. 5th Int. Conf. Informat. Comput. Sci. (ICICoS)*, Nov. 2021, pp. 282–286.
[90] M. Zhang, S. Liu, and B. Zeng, "Hierarchical region proposal refinement network for weakly supervised object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 669–673.
[91] H. Chen and C. Hao, "Mask-Net: A hardware-efficient object detection network with masked region proposals," in *Proc. IEEE 33rd Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Jul. 2022, pp. 131–138.
[92] H. Zhu, M. Xue, Y. Wang, G. Yuan, and X. Li, "Fast visual tracking with Siamese oriented region proposal network," *IEEE Signal Process. Lett.*, vol. 29, pp. 1437–1441, 2022.
[93] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollr, "Learning to refine object segments," in *Proc. 14th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 75–91.
[94] G. Jin, R.-I. Taniguchi, and F. Qu, "Auxiliary detection head for one-stage object detection," *IEEE Access*, vol. 8, pp. 85740–85749, 2020.
[95] L. Song, Y. Li, Z. Jiang, Z. Li, H. Sun, J. Sun, and N. Zheng, "Fine-grained dynamic head for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11131–11141.
[96] S. Miao, R. Feng, and Y. Zhang, "Representation reconstruction head for object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1516–1520.
[97] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788.
[98] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7369–7378.
[99] R. Xia, G. Li, Z. Huang, H. Meng, and Y. Pang, "CBASH: Combined backbone and advanced selection heads with object semantic proposals for weakly supervised object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6502–6514, Oct. 2022.
[100] Y. Shi, J. Wu, S. Zhao, G. Gao, T. Deng, and H. Yan, "Rethinking the detection head configuration for traffic object detection," 2022, *arXiv:2210.03883*.
[101] H. Jiang and Q. Gu, "G-head: Gating head for multi-task learning in one-stage object detection," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.
[102] Y. Zhu, X. Sun, W. Diao, H. Wei, and K. Fu, "DualDA-Net: Dual-head rectification for cross-domain object detection of remote sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–16, 2023, Art. no. 5612616, doi: 10.1109/TGRS.2023.3285747.
[103] G. Pleiss, D. Chen, G. Huang, T. Li, L. van der Maaten, and K. Q. Weinberger, "Memory-efficient implementation of DenseNets," 2017, *arXiv:1707.06990*.
[104] W. Zhang, M. Jiang, and G. Luo, "Evaluating low-memory GEMMs for convolutional neural network inference on FPGAs," in *Proc. IEEE 28th Annu. Int. Symp. Field-Program. Custom Comput. Mach. (FCCM)*, May 2020, pp. 28–32.
[105] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Comput. Archit. News*, vol. 45, no. 1, pp. 615–629, 2017.
[106] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020, doi: 10.1109/ACCESS.2020.2991734.

ARIEF SETYANTO (Member, IEEE) received the Ph.D. degree in electronics and computer science from Essex University, Colchester, U.K., in 2016. He is currently an Associate Professor with the Department of Informatics, Universitas Amikom Yogyakarta, Indonesia. His research interests include image segmentation, image classification, object detection, near-edge computing, and deep learning.

THEOPILUS BAYU SASONGKO (Member, IEEE) received the bachelor's degree in information systems from Universitas Kristen Duta Wacana, in 2012, and the master's degree in electrical engineering from Gadjah Mada University, in 2014. In 2017, he joined Universitas Amikom Yogyakarta, as a Lecturer. His research interests include data mining, artificial intelligence, deep learning, and computer vision.

MUHAMMAD AINUL FIKRI (Member, IEEE) received the bachelor's degree in telecommunication and audio engineering from Andalas University and the master's degree in artificial intelligence and human–computer interaction from Gadjah Mada University. He received a scholarship from the Education Fund Management Agency (LPDP) under the Ministry of Finance of the Republic of Indonesia.

IN KEE KIM (Member, IEEE) received the Ph.D. degree in computer science from the University of Virginia, Charlottesville, VA, USA, in 2018. He is currently an Assistant Professor with the School of Computing, University of Georgia. His research interests include cloud computing, large-scale distributed systems, the IoT/edge computing, and machine learning systems.