A Review of Lightweight Object Detection Algorithms for Mobile Augmented Reality

Mohammed Mansoor Nafea[1], Siok Yee Tan[2]
Center for Artificial Intelligence Technology, Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia, Bangi, 43600 Selangor, Malaysia

Mohammed Ahmed Jubair[3]
Department of Computer Technical Engineering, College of Information Technology, Imam Ja'afar Al-Sadiq University
Al-Muthanna 66002, Iraq

Mustafa Tareq Abd[4]
Department of Computer Technology Engineering, Middle East University College
Baghdad, Iraq

***Abstract*—Augmented Reality (AR) has led to several technologies being at the forefront of innovation and change in every sector and industry. Accelerated advances in Computer Vision (CV), AR, and object detection refined the process of analyzing and comprehending the environment. Object detection has recently drawn a lot of attention as one of the most fundamental and difficult computer vision topics. The traditional object detection techniques are fully computer-based and typically need massive Graphics Processing Unit (GPU) power, while they aren't usually real-time. However, an AR application required real-time superimposed digital data to enable users to improve their field of view. This paper provides a comprehensive review of most of the recent lightweight object detection algorithms that are suitable to be used in AR applications. Four sources including Web of Science, Scopus, IEEE Xplore, and ScienceDirect were included in this review study. A total of ten papers were discussed and analyzed from four perspectives: accuracy, speed, small object detection, and model size. Several interesting challenges are discussed as recommendations for future work in the object detection field.**

***Keywords*—Augmented reality (AR); object detection; computer vision (CV); non-graphics processing unit (Non-GPU); real time**

I. Introduction

As opposed to virtual reality (VR), the term augmented reality (AR) refers to a virtual interface, either in 2D or 3D, that enhances human vision by superimposing additional information (digital content) over the actual surroundings, which results in complete immersion in the imaginary reality[1]. As we can always see and feel the real world around us, it is not possible to get immersed in the virtual world.

AR relies on a device that captures the real world and inverts live animations, virtual objects, texts, data, or sounds observed by the user on a smartphone, tablet, computer, a pair of glasses, or any other on-screen display system [2]. In AR, the virtual information and the real world are synched using geo-localization and embedded sensors (gyroscope, accelerometer) that position the user and adjust the display according to his environment and movements.

Think about driving at night on a mountainous, curved route; also imagine a thick fog covering this area right now. The fog completely hides the trees on either side of the road. Considering how crowded it is, it is challenging to even see the traffic signs from a distance of two to three feet. The fog reflects the headlights' illumination, making driving risky, but stopping could be much riskier. In essence, there is no way to win, Similar cases to this have resulted in numerous multicar accidents. Now imagine being able to view the road's layout on a windshield, displaying the distance between you and the vehicles in front of you, your current speed, exit points, and intersections simply by pushing a button. Even though it's never fully safe to drive in fog, this display can provide the driver access to important information that could help avoid an accident [3].

Again, imagine a new faculty member entering a very large academic library; this new member turns on the camera on his/her cellphone and carefully scans the entire area. Imagine the new member having information about the room he/she is standing on the phone screen. Perhaps the location of the reference books, stacks, and current journals, or where to find help, is recorded by the camera. Imagine a scenario where a new faculty member may have virtual arrows pointing to the book's location., he/she is looking for on display to guide him/her in finding the location of the book. The use of AR makes both scenarios possible [3].

As a method of human-computer interaction, AR merely projects virtual data onto real objects [4]. The accurate superimposition of virtual data onto real objects depends on the detection of real objects and the knowledge of their precise coordinates. Several researchers discussed and elaborated on different challenges faced by AR technology. The following are the main challenges faced by AR technology.

A. Display

One of the main challenges with AR display technology is to create an extensive field of view, high resolution [4], see-through display in a socially suitable shape element. The study by [5] listed some of the challenges in optics and displays that must be addressed, which include offering enough brightness

and sharp display, having a high resolution and extensive area of view, addressing eyestrain, and being in a sunglass-like shape element. Several crucial subjects consist of addressing the AR vergence accommodation challenges, displaying photorealistic content material, and new shape elements which includes contact lenses.

B. Interaction Techniques

This is another challenge which is allowing humans to control AR content material as effortlessly as they do with items within the actual world. One approach that has been explored is utilizing actual gadgets to interact with AR in a method known as Tangible AR [6]. Free-hand gestures are supported by current AR systems; however this will be improved in addition to voice recognition and enabling mixed speech and gesture input in multi-modal interfaces. Future studies might use a variety of different approaches using eye tracking, whole-body input, and various non-verbal indications. Ultimately, [7] recommended more studies on the interaction methods that are not feasible inside the actual world.

C. Social and Ethical Issues

In the long term, the meaningful social and ethical concerns in AR technology are a complicated issue that must be addressed, for instance, identity hacking. However [8] mentioned the privacy implications of seeing individual information in public spaces; the study came up with lists of questionable ethical uses of AR which include deception, surveillance, and behavior modification.

D. Object Detection Techniques

For AR applications like assembly guidance, real-time, scalable object detection is a crucial task. Real-time object detection from RGB images has already been addressed by several Deep Neural Network (DNN) models. Although the majority of current AR and mixed reality (MR) systems can comprehend the 3D geometry of their surroundings, they are unable to recognize and categorize complex items in the real world. Deep Convolutional Neural Networks (CNN) can allow these features; however, it is still challenging to run big networks on mobile devices. It is extremely difficult to offload object detection to the edge or cloud due to the strict requirements for low end-to-end latency and excellent detection accuracy. In [9] proposes an object detector to boost embedded devices' ability to identify objects.

Augmented reality simply displays digital information onto real-world objects as a technique for human-computer interaction. In the era of augmented reality, virtual things created by computers can precisely and instantly overlay physical ones. Predicting bounding boxes and categorizing objects are both steps in the process of object detection which is a key area of computer vision today. Robust detection of objects from natural features of AR is still a complex problem and usually demands high computational time.

DL-based object detection has received significant research interest in current years. For higher picture knowledge techniques, it is vital not only to pay attention to classifying different pictures but to attempt to exactly estimate which objects are present within the pictures and their places (known as object detection) [10]. Many studies have been reported on object recognition using CNN in the field of computer vision.

However, end-to-end DL object identification methods based on regression methods, such as the Single Shot multi-box Detector (SSD) series and the YOLO series, have been successful in detecting objects in real time using GPU-based computers. It is exceedingly difficult to achieve accurate and real-time detection using non-GPU-based PCs and portable devices with low computing capacity because of the high computational needs of many systems. Several researchers discussed that a lot of object detection models have higher computing time which makes their object detection models take a long time to provide the outcomes. The following are the survey's contributions:

1) A comprehensive survey was conducted on the current object detection models for embedded devices published from 2018 to 2022.

2) We summarized numerous object detection methods for mobile and embedded devices.

3) The employment of algorithms to identify their unique and constrained features. We also looked at how these techniques handle problems that arise during the object detection process, as well as the advantages and disadvantages of these techniques for object detection models.

4) Discussion on the current open problems to help direct future studies on improving and enhancing the performance of object detection techniques. The development of the model was done after reviewing many earlier studies from related domains.

There are eight sections in this paper; the introduction and scope of this study are presented in Section I, and the review of the structure of the AR system is presented in Section II. In Section III some applications of object detection were discussed. The most important challenges in the object detection field are presented in Section IV. Previous research on similar topics, the outcome of the analysis, and the limitations and conclusions of such studies are in Sections V, VI, VII, and VIII respectively.

II. AR System

Specific software and hardware are required for the various AR systems, however, the software utilized uses the real-world coordinates via cameras and tracking hardware; the purpose is to convert this location data into an XML file, in software the so-called ARML (Augmented Reality Markup Language) used. The ARML functional blocks establish the relationship between the actual and virtual worlds by identifying the relationship between them; this enables use of virtual items in the real world. These virtual items are controlled by the user's actions that they take [11]. Most devices utilized in AR applications are IoT devices that fall into one of three categories:

- Sensors: Sensors gather information from the real world and send it to an AR app. for example, a mobile device's built-in camera gathers information about a user's environment. Data is processed by the software, which subsequently shows the user predefined content.

With information from cameras or 3D models, the scene's composition is achievable. The tracking device could be (RFID, wireless sensors, accelerometers, GPS, gyroscopes, solid-state compasses, and digital cameras) [11] and have several settings and ranges; they enhance the AR system's tracking accuracy.

- Input devices: These tools let users engage with AR systems. The AR Interface works as a medium between the AR system and these components. The Ikea application's UI could be a good example. The user could move the furniture items in his house by using signs that the program will then interpret as orders [11]. The types of inputs include speaking, blinking, touching, and gesturing, among others. Input devices examples: microphones, touch displays, gesture controllers, styluses, and pointers [12].
- Output devices: Users can interact with the AR system using these tools. Whenever these tools are used for a particular purpose, they are often worn on the user's forehead. HMDs, monitors, and wearable technology are some examples of these devices [13].

Offering a solution in real-time is one of an AR system's main characteristics. The user experience is not the only benefit of AR technology. t offers excellent economic opportunities for service providers and companies [13]. As AR has been more widely used, it allows more e-production of wireless networks, sensors, high-end cameras, cellphones, and other devices, but there are several factors to take into account while designing an AR system and the architecture that supports it. Such as the high quality and how immersive AR is, also the monitoring and rendering slowness [11]. There are some limitations related to the AR systems such as:

A. Hardware Limitations

A variety of smart devices are used to create AR systems. From the least powerful to the most powerful devices that work across multiple environments. Therefore, one of the key goals is to lower these devices' energy consumption to increase their efficiency[14]. In addition to the energy consumption, the cost of the AR devices may easily cost thousands of dollars which makes them unreachable to everyone except the most committed pioneers or early adopters.

B. Software Limitations

The most efficient hardware is useless if it doesn't come with incredible software. Although the architecture of such AR systems has advanced significantly, the software component for such a complex system still has certain challenges to overcome and objectives to achieve [11]. The flexible operating system has to be coded, sized, and powered more efficiently. The following are some examples of operating systems that can be used in the context of AR (FreeRTOS, OpenWSN, and TynyOS). Additionally, there are dedicated browsers for AR, including Mozilla's "Firefox Reality." And several more devices are still under development. Therefore, it is clear that it is needed to design suitable toolkits having the support of various devices and applications across numerous platforms.

C. Lack of Privacy

The perceived risk associated with AR is one of the key privacy concerns. A person's privacy is at risk since AR technology can monitor what they are doing and collects a ton of data about them and their activities.

D. Devices Compatibility

In AR systems, there should be no issues with communication between objects and devices, the compatibility of such devices is one of the main issues with an AR system's design [15]. To address this issue, we should focus on “We must improve semantic exchange between objects and devices and between devices with each other. The semantic web may be used to do this, which can improve the quality of digital content seen via the user interface.”

E. User Intervention

The dependency of AR apps on the user and their activity is another issue. Any IoT device must be independent and sensitive [11]. Its role needs to be less obvious to the user so that consumers may have a system that is much more robust, even when there are problems [15]. Fig. 1 presents the components of AR system.

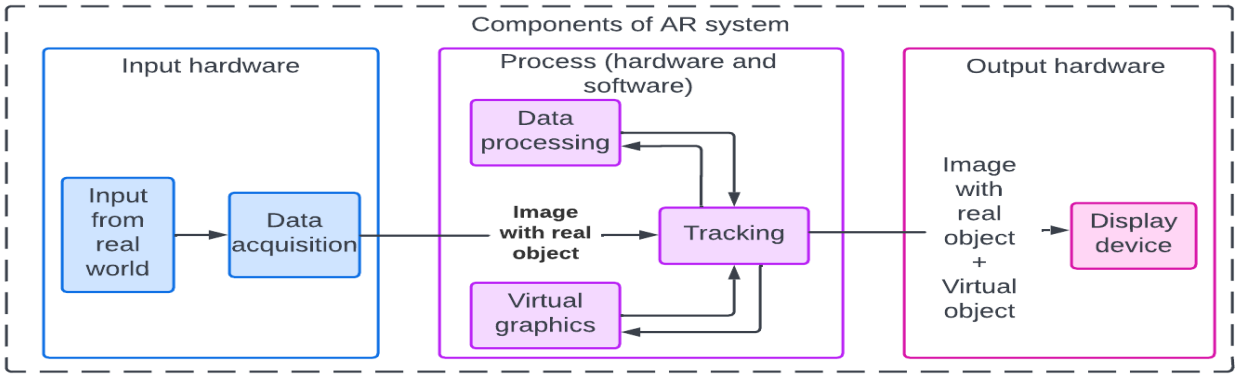

Fig. 1. AR System Components.

III. Applications of Object Detection

This section will discuss some recent significant detection applications, such as the detection of pedestrians, faces, traffic signs and light, and texts.

A. Pedestrian Detection

Nowadays, pedestrian detection has received extensive research, which is closely related to pedestrian tracking person re-identification. Before the recent development in DCNN algorithms, some studies combined boosted decision forests with hand-crafted features to develop pedestrian detection methods. also, to address the deformation and occlusion, part-based methods and explicit handling of occlusion are problematic [16].

B. Face Detection

Face detection is vital for numerous face applications and behaves as an essential face recognition pre-processing step. One of the initial computer vision technologies that has successfully supported object detection is face detection, and many of its amazing principles are still having a big impact on object detecting technology today. Face detection is currently used in all parts of life, for digital cameras' "smile" detection, “face swiping” in online retail, face makeup, etc. [16].

C. Text Detection

The issues with text detection in video and images have attracted more notice recently, as shown by the appearance of recent” Robust Reading” competitions in 2003, 2005, 2011, and 2013. There are several main causes for this trend, including a rising number of applications. Text is one of the most artistic methods of communication and can be integrated into documents or scenes as a technique of connecting information. The same problems with computer vision and pattern recognition are connected to text identification in photographs caused by low quality or worse, despite the fact that many studies consider OCR technology (Optical Character Recognition) to be a settled subject. Text detection field still has a big space for search [17].

D. Traffic Sign and Traffic Light Detection

With the advancement of autonomous vehicle technologies, the detection of traffic signs and lights has received a lot of attention nowadays [16]. Even though the computer vision community has mostly highlighted, Traffic Light Recognition has now made a name for itself as a standard basis for pedestrian and general object detection, until 2010, the majority of traffic sign/light detection technologies, except a very small number of works, did not adhere to this paradigm.

IV. Challenges of Object Detection

Finding and identifying various instances of videos and pictures is made easier with the use of a type of computer vision technology called object detection. The most typical object detection problems that data scientists should be aware of are listed below:

A. Small Object Detection

Small objects are typically difficult to detect because of their noisy depiction and low resolution. Currently used object identification pipelines often find small objects by learning the representations of every object at different scales. Typically, such ad hoc systems only provide performance improvements sufficient to offset their computing costs. Numerous real-world applications, such as pedestrian detection and traffic sign identification for improved autonomous driving, frequently involve small objects. However, the detection of small objects is more difficult than that of regular objects, and effective solutions are yet to be established [18]. Even though some of these problems with small object detection have received some attention, most of the proposed methods erroneously increase the feature dimension or enrich the data to improve the effectiveness of small object detection [19].

B. Object Detection Model Size

The present CNN frameworks are deepening and becoming more complex, despite the fact that the accuracy of such network architectures may match or exceed that of human vision, they frequently need extremely high amounts of computer power [20]. Fast object detection techniques have advanced significantly; however, it is still difficult to apply CNN architectures on non-GPU or mobile devices. The usage of CNN-based technologies has brought rise to major research topics in real-time and lightweight network models [21] [22] for object detection in mobile devices. This is driven by the rapid advancement in mobile and embedded intelligent gadgets with low power consumption and limited computing power, such as AR glasses and small intelligent unmanned aerial vehicles (UAVs).

C. Speed

Deep neural network-based methods have frequently performed better than other methods in object detection evaluations. There are two categories of these models - those with Regional Proposal Network (RPN layer) and those without RPN; those with the RPN layer are normally the faster R-CNN & R-FCN. YOLO and SSD are the models without it. YOLO and SSD are also called single-shot detection models. In general, the models with Region Proposal Network are more accurate while the models without Region Proposal Network and SSD are faster. The size of the model could affect the speed of detection so the pruning of model layers will speed up the model [23] [24]. The best models are the models that balance accuracy and speed.

D. Viewpoint Variation

Deep CNNs are only capable of modeling 2D transformation fields using the existing methods for encoding spatial invariance. This fact does not consider that objects in a 2D space are a projection of the ones in 3D and hence, have a limited power to severely shift object viewpoints is not taken into consideration by this [25]. Recently, CNN-based joint object recognition and viewpoint estimate has drawn attention as a potential solution for viewpoint variations. Before predicting the relative stiff transformation between each picture's 2D coordinate and the camera point in 3D space, must first identify the location and kind of objects in an image viewpoint. Estimation, and category classification problems are intrinsically incongruent.

E. Deformation

Sometimes interesting items may be flexible and distorted in unusual ways. Although a person can be detected by an object detector in a different situations, because of his twisted orientations, it will be difficult for the object detector to identify the same person.

The most advanced item detection technology currently available uses Deformable Parts Models (DPM). They do not appear to be the best at representing deformations. The deformations of objects are frequently continuous and not limited to large parts. Variations in popular and efficient object detectors are due to changes in appearance and deformations[26].

F. Occlusion

In real-world images and videos, occlusion is a frequent issue that presents a significant challenge for object detection. For instance, 70% of pedestrians in the Caltech Pedestrian Dataset, are obscured in at least one video sequence frame, and 19% are obscured in every frame. Almost 50% of these pedestrian occlusions were classified as heavy [27]. According to Dollar et al., even with minor occlusion, the performance of typical approaches to detect objects decreases significantly, and with heavy occlusion, it falls even further. Therefore, better performances are provided by object detectors that can learn and infer visible patterns by focus on fix the occlusion issue [28].

G. Shadow and Illumination Conditions

The alteration of illumination is a typical issue with object detection processes. Some techniques are offered to reduce the effect of changes in illumination and shadow caused by moving objects; an example of such techniques is the Moving Object Detection and Shadow Removing under Changing Illumination Condition model. The shadow causes several problems with object localization, segmentation, recognition, and tracking. The shadow may also result in the objects merging, it can cause the shapes of the objects to change, might even result in things going missing and lead to the foreground being confused for the background. It is equally challenging to capture clear moving objects when illumination changes because of the chances of mistaking some background pixels for foreground pixels. The effectiveness of subsequent procedures (such as tracking, recognition, classification, and activity analysis) that require precise moving object detection and accurate acquisition of its exact shape, therefore there is great impact by removing shadows and controlling illumination fluctuations [29, 30].

III. Previous Work

Numerous theoretical and empirical research has covered and elaborated on AR. This study focuses on approaches and procedures that can be utilized to enhance deep learning AR models. The following are previous studies about AR that were conducted by various studies:

In [31], Trident-YOLO, an upgraded version of the YOLOv4-tiny network with better accuracy and real-time speed was suggested. The most significant improvements made by this model are to the set of tools indicated by Alexey and the suggestions of CSP-RFBs and CSP-SPPs, which are appropriate for thin object detection networks. In order to increase the accuracy and recall of lightweight object detection, the network topology was redesigned, and a trident feature pyramid network (Trident-FPN) was suggested by the authors. This Trident-FPN produces a multi-scale feature map of the model while only slightly increasing the computational cost in terms of floating-point operations per second (FLOPs).

The study of [32] introduced the TRC-YOLO, TRC-YOLO proposed the pruning of the YOLOv4-tiny convolution kernel and the addition of an expansive convolution layer to the residual network model to produce an hourglass-shaped Cross-Stage partial Present (CSP) structure. The introduction of TRC-YOLO enhanced the mAP and real-time speed while minimizing the model size which was achieved by minimizing the number of YOLOv4-tiny model parameters. The CSP Res Net module was then enhanced and integrated to boost the model's capacity for feature extraction. In order to obtain higher quality feature images and to enable the model to concentrate on important feature areas and channels, the RFBs module was included to this model.

A mobile inverted bottleneck module is used as the foundation of the feasible and lightweight object detection model presented by [33]; the model was based on deep CNN. Additionally, an improved spatial pyramid pooling was used to concatenate the multi-scale local region characteristics to increase the network's receptive field. The testing results on the aerial picture datasets VEDAI and VisDrone show that the enhanced YOLOv4-tiny model performs significantly better than the original YOLOv4-tiny model. For the VEDAI and VisDrone datasets, the suggested model outperformed the results with mAP of 53.11 percent and 24.73 percent, respectively.

The study by [34] suggests embedded YOLO to enhance the efficiency of low-level features; the study initially suggested a new backbone network topology called the ASU-SPP network but later developed a more straightforward version of the neck network module PANet-Tiny to make computations simpler. Finally, depth-wise separable convolution was employed in the detecting head module to minimize the number of convolution stacks. The embedded YOLO model was compared with the conventional lightweight model after being verified by the COCO test dataset and online tests; it was discovered that the mAP performance was preserved.

In [21], a newly developed lightweight CSL-Module was presented; the new approach showed a comparative performance with previous approaches of a similar nature but due to limited computing resources, two additional components (CSL-Bone and CSL-FPN) were proposed to achieve superior performance with fewer FLOPs. However, achieving low computation cost depends on how the redundant features are generated as the CSL-Module can lower computation costs considerably. Research done at MS-COCO demonstrates that the suggested CSL-Module can approximate the fitting ability of Convolution-3x3.

A real-time object detection approach for non-GPU systems was proposed by [35] in order to help users of low-

configuration computers. Real-time object detection on CPU-based computers is now possible thanks to the optimization of YOLO with OpenCV for CPU-based computing. On some non-GPU machines, the CPU-based YOLO model can detect objects from videos with an accuracy of 80–99 percent and frame rates between 10.12 and 16.29. Comparison with other GPU-based frameworks showed that the proposed model is suitable for CPU-based applications because CPU Based YOLO obtains 31.05 percent mAP.

A study by [35] suggested a brand-new, DL-based lightweight object detection technique. Based on YOLOv4, the study proposed YOLOv4-tiny is proposed to simplify the network topology and minimize parameters, making it appropriate for development on mobile and embedded devices. The proposed approach achieved faster object detection compared to YOLOv4-tiny and YOLOv3-tiny as evidenced by the simulation results; it also achieved an almost similar mean value of average precision as the YOLOv4-tiny. The authors also proposed two identical ResBlock-D modules for the replacement of two CSPBlock modules in the YOLOv4-tiny network to reduce the object detection process. An auxiliary network block that employs two 3x3 convolutions networks, spatial attention, concatenate operation, and channel attention was also proposed for the global feature extraction in order to balance the object detection time and accuracy.

Mixed YOLOv3-LITE was developed by [22] as a mobile and non-GPU compatible lightweight real-time object detection network. The proposed approach supplements the ResBlocks and parallel high-to-low resolution subnetworks that are YOLO-LITE-based. The detector was developed using narrower and shallower convolutional layers compared to those in YOLOv3; this reduces the required level of computation and number of parameters to be trained, hereby improving the network operation speed. These considerations were aimed at solving the problems of limited computing power and excess power consumption in mobile and embedded smart devices.

In [23], a real-time object detection paradigm was developed for mobile devices like laptops or phones without a GPU. The study YOLOLITE to provide a smaller, quicker, and more effective model based on the original YOLOV2 algorithm; the introduction of YOLOLITE increased the accessibility of real-time object identification to a variety of devices. With its success in bringing object detection to non-GPU machines, YOLO-LITE demonstrates the enormous potential of shallow networks for lightweight real-time object detection networks. For such a modest system, running at 21 FPS on a non-GPU computer is quite encouraging, and it demonstrates why batch normalization must be queried for smaller shallow networks. It showcased the capability of shallow networks in fast non-GPU object detection devices. It also proves that shallow networks do not necessarily require batch normalization but reduce the network's overall performance. To sum up, the YOLO-LITE detector is composed of several parts:

1) Input: Image, Patches, Image Pyramid

2) Backbone: Darknet-53

3) Heads: Dense Prediction: one-stage.

A YOLObile framework that relies on compression-compilation co-design was designed in [36] to enable real-time object recognition on mobile devices. The study proposed a brand-new block-punched pruning approach for any kernel size. Furthermore, advanced compiler-assisted optimizations in conjunction with a GPU-CPU collaborative strategy was also suggested to increase computational efficiency on mobile devices. According to experimental findings, the pruning strategy successfully compressed YOLOv4 at a 14 rate with 49.0 mAP. On the Samsung Galaxy S20, a 17 FPS inference performance was reached using the proposed YOLObile framework with the GPU-CPU collaboration strategy. The proposed YOLObile also includes a mobile GPU-CPU collaborative computation strategy to increase the computational effectiveness of DNNs on mobile gadgets. The evaluation shows that the proposed YOLObile framework achieved high hardware parallelism and excellent accuracy. Fig. 2 summarized the percentage of previous work studies that focus on improve each attribute.

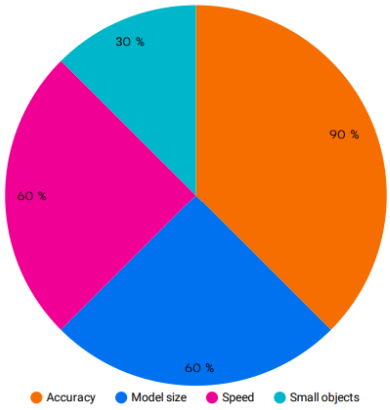

Fig. 2. Applied Approaches.

V. Outcome and Findings

The main finding of this paper is that CNNs object detection models are through an advanced transformation in the field of computer vision. The development of object detection models to be appropriate even for mobile and devices without GPU, which offers customized models and small-scale proof-of-concept studies, is what highlights this transformation. These models' studies provide a foundation for large-scale scientific research issues that employ CNNs. As a result of this tendency, CNNs object detection models will eventually be able to compile a worldwide, digital inventory of everything in order to study object dynamics and their effects on human activity. This study analyzes 10 articles to give a thorough overview of the convolutional neural network (CNN) for object detection in general. Another finding is that always there is a trade-off between speed and accuracy. When focus on increasing accuracy this will make the model more complex which will slow the model down.

TABLE I. SUMMARY OF PREVIOUS WORK

Reference	Method	Summary	Evaluation Metrics	Dataset Used	Limitations
[31]	Trident-YOLO	Introduce an object detection network that is designed for mobile devices with limited computing power	FPS mAP BFLOPs	PASCAL VOC MS COCO	There is still room for improvement in the detection of complex scenes.
[32]	TRC-YOLO	Proposed a lightweight object detection model based on yolov4-tiny.	FPS mAP	PASCAL VOC MS COCO	Model speed needs to be higher
[33]	Automated object detection on aerial images for limited capacity embedded device	Feasible and lightweight object detection model based on deep CNN	FPS mAP input size F1-score IoU FLOPs	VisDrone VEDAI	The model was based on yolov4 tiny and slower than yolov4 tiny.
[34]	Embedded YOLO	Proposed an ultralightweight target detection network model	FPS mAP parameters Size Latency	MS-COCO	The model's map needs to improve.
[21]	CSLYOLO	Proposed a new lightweight convolution method CSL-Module and proposed two components CSL-Bone and CSL-FPN, both of which achieve better performance with fewer flops.	FPS mAP Params FLOPs	MS-COCO	The model's running speed needs to improve
[42]	CPU Based YOLO	Optimized YOLO with OpenCV to enable real-time object detection on CPU-based devices.	FPS mAP	COCO	Model speed needs to be higher
[24]	YOLOv4-tiny	Proposed Yolov4-tiny based on Yolov4 to reduce the network structure and ensure suitability with mobile and embedded devices.	FPS mAP	MS COCO	Still not implementable for non-GPU devices
[22]	Mixed YOLOv3-LITE	Proposed "an efficient lightweight object detection network that uses a shallow-layer, narrow-channel, and multi-scale feature image parallel fusion" structure	FPS mAP Params GFLOPs Precision Recall F1	PASCAL VOC 2007 & 2012	Need to improve the map
[23]	YOLO-LITE	The proposed detector model runs at "about 21 FPS on a non-GPU computer and 10 FPS after implemented onto a website with only 7 layers and 482 million" FLOPS.	FPS mAP Layers FLOPS Loss	PASCAL VOC 2007 and 2012 COCO 2014	Need to improve the map
[36]	YOLObile	Proposed yolobile framework, a real-time object detection on mobile devices via compression-compilation co-design.	Input Size Backbone Weights FLOPs mAP FPS	COCO dataset	Need to improve detection speed

A. Applied Approaches

The approaches utilized to address the difficulties and resolve the object detection problems are covered in depth in Section V. Tables I and II summarized each of the publications reviewed and how they addressed the difficulties faced by the researchers. As seen in Table II and Fig. 1, most research focus on improving accuracy (90 percent), handling speed (60 percent) have received far more attention than any other problems.

B. Investigated Datasets

Datasets are significant motivators for specific applications and crucial for the development of deep-learning algorithms. The number of case studies and datasets considered has proven to be challenging. It is advantageous and necessary to use a number of datasets, each of which supports a range of parameters and a predetermined composition problem, in order to assess the efficiency of the suggested algorithms. There are many object detection datasets that are readily available in the research domain: COCO [37] [23] [24] [35, 36] http://cocodataset.org/ and PASCAL VOC [23] [38-41] http://host.robots.ox.ac. uk/pascal/VOC/., cifar-10 [42-46]

https://www.cs.toronto.edu/~kriz/cifar.html, bdd100k dataset [47-51] https://www.bdd100k.com/, LISA Traffic Sign Detection Dataset[52-56], KITTI [49, 57-60] http://www.cvlibs.net/datasets/kitti/, SUN-RGB-D [61-65]https://rgbd.cs.princeton.edu/,nuScenes[66-70] https://www.nuscenes.org/, Visual genome [71-75] https://visualgenome.org/ , MPII [76-80] http://human-pose.mpi-inf.mpg.de/ , and imagine [81-85] https://image-net.org/index.php. Some researchers have rarely relied on datasets that are generated synthetically.

VI. Limitations of Object Detection in Applied Approaches

This section discusses a variety of essential object detection techniques as well as object detection issues and constraints. The main problems with object detection systems are their speed, accuracy, difficulty in detecting small objects, and model size. Numerous algorithms have been developed to address these issues, but none of them are completely successful. The enumerated algorithms provide the following solutions to the problems.

- Accuracy

A common metric used to assess the accuracy of object detection models is Mean Average Precision (mAP). The mAP is calculated by comparing the ground-truth bounding box to the detected box; higher mAP scores imply that the associated models have higher detection accuracies. The accuracy metric is the most important factor to assess any object detector. The new models used improved techniques in Neck part like (FPN, PANet, and SPP) and the Residual Blocks in Backbone part also using advanced Head part like head in YOLOv4 to increase the accuracy of bounding boxes position. All these techniques and more are utilized to improve the accuracy of detectors like [23, 24, 36].

- Speed

Beside the accuracy factor, detector speed should be taken in account. The effectiveness of any intelligent system and AR gadgets depends on an effective and quick object detection algorithm. Without real-time detection the model will be useless in field of autonomous vehicles and many other fields. To tackle this issue many researchers suggested new features for example (simple the network structure, reduce parameters, and using simple parts instead of complex ones) in YOLOv4-tiny the author used FPN without using SPP which used in YOLOv4, additionally YOLO-LIE model minimize YOLOv2 by using less number of backbone layers. However, A constant trade-off always exists between speed and accuracy in larger models [23, 36].

- Small object detection

The detection of small objects is a difficult computer vision task and has been widely used in defense, military, transportation, industry, etc. to improve understanding of small object detection. Object detection has recently made tremendous strides but despite these advancements, there is still significant variation in performance between the detection of large and small objects. To solve the issue of small object detection, some models have been developed, such as those developed in [21, 22, 36] in [21] CSL-YOLO model, because of the huge number of small objects in MS-COCO dataset the author improved new version of feature pyramid network called (CSL-FPN) in this Network Before K-means, a scale limit like Eq. (3) has been included so that the distribution of anchors produced is more in line with the scale of each output layer. In the results, using CSL-FPN has increase the accuracy of detect small objects. However, small objects detection still needs more studies and it's an open issue need more efforts to be improved.

- Model size

Object detection model size has many effects on the speed and accuracy, the trade-off between accuracy in larger models and speed in lightweight versions is continual. The simple network structure and fewer parameters make it ideal for mobile and embedded devices. On other hand the complex structure and more parameters may be caused to improve model accuracy. Modern object detection techniques used in cars now rely significantly on the sensor output from costly radars & depth sensors that make them unsuitable for usage in everyday situations. However, this increase in accuracy may not be useful to address the problem in many real-world applications that demand real-time. performance carried out on a platform with restricted computational resources [22-24, 35]. Simplicity of detector is an open issue due to need of lightweight models in mobile and non-GPU devices. In Table II we can see the summary of improved attributes for 10 studies.

TABLE II. Improved Attributes

Author	Year	Method	Speed	Accuracy	Model Size	Small objects
Wang, et al., [31]	2022	Trident-YOLO	✓	✓		
Wang, et al., [32]	2021	TRC-YOLO	✓	✓		
Junos et al., [33]	2022	Automated object detection on aerial images for limited capacity embedded device			✓	✓
Feng et al., [34]	2021	Embedded YOLO	✓	✓		
Zhang et al., [21]	2021	CSLYOLO		✓		✓
Ullah et al., [35]	2020	CPU Based YOLO		✓	✓	
Jiang et al., [24]	2020	YOLOv4-tiny	✓	✓	✓	
Zhao et al., [22]	2020	Mixed YOLOv3-LITE	✓	✓	✓	✓
Huang and Chen [23]	2020	YOLO-LITE	✓	✓	✓	
Cai et al., [36]	2020	YOLObile		✓	✓	

VII. CONCLUSION

The study focused on CNN-based light object detection in the field of AR, and it introduces the CNN's structure, the CNN-based object detection framework, and several techniques for enhancing detection performance. Surveys on the application domains, model components, experience metrics, used datasets, and model performance of lightweight object detection models were also conducted; the challenging problems in the field of AR were also highlighted. Nevertheless, there are still several technical and application-related issues in AR. The performance of CNN in terms of real-time, accuracy, and adaptability was better than that of the traditional approaches, but there is still much potential for improvement. Enhancing the object detection algorithm's structure can minimize the loss of feature information while fully leveraging the relationships between the object and the context. Numerous studies have already been done to address non-GPU and embedded devices. Although the improved methods performed better than the conventional methods, the accuracy still needs to be improved to handle a complex environment. A well-established method specifically for addressing the issue of small object detection is yet to be achieved, however, improvisation in small object methods allows for the achievement of acceptable accuracy values, though it is vulnerable to additional processing time. New methods can be developed in the future by leveraging the strength of the recent trends for improved performance. The current methodologies, for instance, can be improved by hybridizing non-GPU approaches and small object detection with GPU-based approaches.

ACKNOWLEDGMENT

This work was supported by the Universiti Kebangsaan Malaysia with Grant Numbers: PDI-2021-026 and GUP-2020-060.

REFERENCES

[1] Chiang, F.-K., X. Shang, and L. Qiao, Augmented reality in vocational training: A systematic review of research and applications. Computers in Human Behavior, 2022. 129: p. 107125.

[2] Tan, S.Y., H. Arshad, and A. Abdullah, An improved colour binary descriptor algorithm for mobile augmented reality. Virtual Reality, 2021. 25(4): p. 1193-1219.

[3] Gao, X., et al., Effects of Augmented-Reality-Based Assisting Interfaces on Drivers' Object-wise Situational Awareness in Highly Autonomous Vehicles. arXiv preprint arXiv:2206.02332, 2022.

[4] Scavarelli, A., A. Arya, and R.J. Teather, Virtual reality and augmented reality in social learning spaces: a literature review. Virtual Reality, 2021. 25(1): p. 257-277.

[5] Lee, Y.-H., T. Zhan, and S.-T. Wu, Prospects and challenges in augmented reality displays. Virtual Real. Intell. Hardw., 2019. 1(1): p. 10-20.

[6] Nizam, S.M., et al., A review of multimodal interaction technique in augmented reality environment. Int. J. Adv. Sci. Eng. Inf. Technol, 2018. 8(4-2): p. 1460.

[7] Goh, E.S., M.S. Sunar, and A.W. Ismail, 3D object manipulation techniques in handheld mobile augmented reality interface: A review. IEEE Access, 2019. 7: p. 40581-40601.

[8] Slater, M., et al., The ethics of realism in virtual and augmented reality. Frontiers in Virtual Reality, 2020. 1: p. 1.

[9] Liu, L., H. Li, and M. Gruteser. Edge assisted real-time object detection for mobile augmented reality. in The 25th annual international conference on mobile computing and networking. 2019.

[10] Li, X., et al. Object detection in the context of mobile augmented reality. in 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). 2020. IEEE.

[11] Arena, F., et al., An Overview of Augmented Reality. Computers, 2022. 11(2): p. 28.

[12] Viglialoro, R.M., et al., Review of the augmented reality systems for shoulder rehabilitation. Information, 2019. 10(5): p. 154.

[13] Ping, J., Y. Liu, and D. Weng. Comparison in depth perception between virtual reality and augmented reality systems. in 2019 ieee conference on virtual reality and 3d user interfaces (vr). 2019. IEEE.

[14] Jeffri, N.F.S. and D.R.A. Rambli, A review of augmented reality systems and their effects on mental workload and task performance. Heliyon, 2021. 7(3): p. e06277.

[15] Solbiati, M., et al., Augmented reality for interventional oncology: proof-of-concept study of a novel high-end guidance system platform. European radiology experimental, 2018. 2(1): p. 1-9.

[16] Zhao, Z.-Q., et al., Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 2019. 30(11): p. 3212-3232.

[17] Ye, Q. and D. Doermann, Text detection and recognition in imagery: A survey. IEEE transactions on pattern analysis and machine intelligence, 2014. 37(7): p. 1480-1500.

[18] Kisantal, M., et al., Augmentation for small object detection. arXiv preprint arXiv:1902.07296, 2019.

[19] Liu, Y., et al., A survey of research and application of small object detection based on deep learning. Acta Electonica Sinica, 2020. 48(3): p. 590.

[20] Zaidi, S.S.A., et al., A survey of modern deep learning based object detection models. Digital Signal Processing, 2022: p. 103514.

[21] Zhang, Y.-M., et al., CSL-YOLO: A New Lightweight Object Detection System for Edge Computing. arXiv preprint arXiv:2107.04829, 2021.

[22] Zhao, H., et al., Mixed YOLOv3-LITE: a lightweight real-time object detection method. Sensors, 2020. 20(7): p. 1861.

[23] Huang, R., J. Pedoeem, and C. Chen. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. in 2018 IEEE International Conference on Big Data (Big Data). 2018. IEEE.

[24] Jiang, Z., et al., Real-time object detection method based on improved YOLOv4-tiny. arXiv preprint arXiv:2011.04244, 2020.

[25] Wang, X., K. Wang, and S. Lian. Deep consistent illumination in augmented reality. in 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). 2019. IEEE.

[26] Mordan, T., et al., End-to-end learning of latent deformable part-based representations for object detection. International Journal of Computer Vision, 2019. 127(11): p. 1659-1679.

[27] Hebborn, A.K., N. Höhner, and S. Müller. Occlusion matting: realistic occlusion handling for augmented reality applications. in 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). 2017. IEEE.

[28] Wang, A., et al. Robust object detection under occlusion with context-aware compositionalnets. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[29] Shakeri, M. and H. Zhang. Moving object detection under discontinuous change in illumination using tensor low-rank and invariant sparse decomposition. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[30] Tan, S.Y., H. Arshad, and A. Abdullah, A new illumination invariant feature based on freak descriptor in RGB color space. Journal of Theoretical and Applied Information Technology, 2016. 93(1): p. 207.

[31] Wang, G., et al., Trident‐YOLO: Improving the precision and speed of mobile device object detection. IET Image Processing, 2022. 16(1): p. 145-157.

[32] Wang, G., et al., TRC‐YOLO: A real‐time detection method for lightweight targets based on mobile devices. IET Computer Vision, 2021.

[33] Junos, M.H., A.S.M. Khairuddin, and M. Dahari, Automated object detection on aerial images for limited capacity embedded device using a

lightweight CNN model. Alexandria Engineering Journal, 2022. 61(8): p. 6023-6041.

[34] Feng, W., et al., Embedded YOLO: A Real-Time Object Detector for Small Intelligent Trajectory Cars. Mathematical Problems in Engineering, 2021. 2021.

[35] Ullah, M.B. CPU Based YOLO: A Real Time Object Detection Algorithm. in 2020 IEEE Region 10 Symposium (TENSYMP). 2020. IEEE.

[36] Cai, Y., et al., Yolobile: Real-time object detection on mobile devices via compression-compilation co-design. arXiv preprint arXiv:2009.05697, 2020.

[37] Lin, T.-Y., et al. Microsoft coco: Common objects in context. in European conference on computer vision. 2014. Springer.

[38] Redmon, J. and A. Farhadi. YOLO9000: better, faster, stronger. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[39] Wang, X., A. Shrivastava, and A. Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[40] Redmon, J., et al. You only look once: Unified, real-time object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[41] Ren, S., et al., Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015. 28.

[42] Salimans, T., et al., Improved techniques for training gans. Advances in neural information processing systems, 2016. 29.

[43] He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[44] Zoph, B., et al. Learning transferable architectures for scalable image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[45] Chauhan, K., et al. Robust outlier detection by de-biasing VAE likelihoods. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[46] Huang, G., et al., Multi-scale dense networks for resource efficient image classification. arXiv preprint arXiv:1703.09844, 2017.

[47] Wang, J., et al., Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 2020. 43(10): p. 3349-3364.

[48] Wu, D., et al., Yolop: You only look once for panoptic driving perception. arXiv preprint arXiv:2108.11250, 2021.

[49] Cai, Z. and N. Vasconcelos, Cascade R-CNN: high quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence, 2019. 43(5): p. 1483-1498.

[50] Hou, Y., et al. Learning lightweight lane detection cnns by self attention distillation. in Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[51] Tran, L.-A., et al., Robustness Enhancement of Object Detection in Advanced Driver Assistance Systems (ADAS). arXiv preprint arXiv:2105.01580, 2021.

[52] Møgelmose, A., D. Liu, and M.M. Trivedi. Traffic sign detection for us roads: Remaining challenges and a case for tracking. in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). 2014. IEEE.

[53] Møgelmose, A., D. Liu, and M.M. Trivedi, Detection of US traffic signs. IEEE Transactions on Intelligent Transportation Systems, 2015. 16(6): p. 3116-3125.

[54] Lopez-Montiel, M., et al. Evaluation of algorithms for traffic sign detection. in Optics and Photonics for Information Processing XIII. 2019. SPIE.

[55] Zhang, J., et al., A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. IEEE access, 2020. 8: p. 29742-29754.

[56] Brown, W.S., K. Roy, and X. Yuan. US Traffic Sign Recognition Using CNNs. in Proceedings of SAI Intelligent Systems Conference. 2020. Springer.

[57] Chen, S., et al. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. in Chinese Conference on Biometric Recognition. 2018. Springer.

[58] Shi, S., et al. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[59] Zhu, B., et al., Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492, 2019.

[60] Simon, M., et al. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019.

[61] Qi, C.R., et al. Deep hough voting for 3d object detection in point clouds. in proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[62] Misra, I., R. Girdhar, and A. Joulin. An end-to-end transformer model for 3d object detection. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[63] Xie, Q., et al. Mlcvnet: Multi-level context votenet for 3d object detection. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[64] Zhao, X., et al. Object detection with a unified label space from multiple datasets. in European Conference on Computer Vision. 2020. Springer.

[65] Huang, S., et al., Perspectivenet: 3d object detection from a single rgb image via perspective points. Advances in neural information processing systems, 2019. 32.

[66] Stäcker, L., et al. Deployment of Deep Neural Networks for Object Detection on Edge AI Devices with Runtime Optimization. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[67] Yang, Z., et al. 3d-man: 3d multi-frame attention network for object detection. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[68] Han, J., et al., SODA10M: A Large-Scale 2D Self/Semi-Supervised Object Detection Dataset for Autonomous Driving. arXiv preprint arXiv:2106.11118, 2021.

[69] Chen, Q., S. Vora, and O. Beijbom, PolarStream: Streaming Lidar Object Detection and Segmentation with Polar Pillars. arXiv preprint arXiv:2106.07545, 2021.

[70] Zhou, Y., et al. Monocular 3d object detection: An extrinsic parameter free approach. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[71] Kehat, G. and J. Pustejovsky. Neural Metaphor Detection with Visibility Embeddings. in Proceedings of* SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics. 2021.

[72] Bosch, M., et al., Contextual Sense Making by Fusing Scene Classification, Detections, and Events in Full Motion Video. arXiv preprint arXiv:2001.05979, 2020.

[73] Zhou, L., et al., Object relation detection based on one-shot learning. arXiv preprint arXiv:1807.05857, 2018.

[74] Schwarz, M., et al., RGB-D object detection and semantic segmentation for autonomous manipulation in clutter. The International Journal of Robotics Research, 2018. 37(4-5): p. 437-451.

[75] Yu, R., et al. Visual relationship detection with internal and external linguistic knowledge distillation. in Proceedings of the IEEE international conference on computer vision. 2017.

[76] Simon, T., et al. Hand keypoint detection in single images using multiview bootstrapping. in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017.

[77] Rogez, G., P. Weinzaepfel, and C. Schmid, Lcr-net++: Multi-person 2d and 3d pose detection in natural images. IEEE transactions on pattern analysis and machine intelligence, 2019. 42(5): p. 1146-1161.

[78] Luvizon, D.C., H. Tabia, and D. Picard, Human pose regression by combining indirect part detection and contextual information. Computers & Graphics, 2019. 85: p. 15-22.

[79] Sumer, O., T. Dencker, and B. Ommer. Self-supervised learning of pose embeddings from spatiotemporal relations in videos. in Proceedings of the IEEE International Conference on Computer Vision. 2017.

[80] Sekii, T. Pose proposal networks. in Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[81] Simonyan, K. and A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[82] Chen, B.-C., et al. Efficient object embedding for spliced image retrieval. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[83] Gupta, V., et al., Deep learning-based automatic detection of poorly positioned mammograms to minimize patient return visits for repeat imaging: A real-world application. arXiv preprint arXiv:2009.13580, 2020.

[84] Gao, Y., et al., Utilizing the instability in weakly supervised object detection. arXiv preprint arXiv:1906.06023, 2019.

[85] Joung, S., et al. Cylindrical convolutional networks for joint object detection and viewpoint estimation. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.