ENHANCING OBJECT DETECTION IN MOBILE AUGMENTED REALITY: A NOVEL FRAMEWORK INTEGRATING KNOWLEDGE DISTILLATION AND UNSUPERVISED DOMAIN ADAPTATION

Xiangyun Zeng[1], Siok Yee Tan [1*],Mohammad Faidzul Nasrudin[1], Mohammad Kamrul Hasan[2*]

[1]Center for Artificial Intelligence Technology, Faculty Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600 Malaysia.
[2] Faculty Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600 Malaysia.

ARTICLE INFO

***Article history*:**
Received: 20.08.2024.
Received in revised form: 19.03.2025.
Accepted: 19.03.2025.

Keywords:
Knowledge distillation
Mobile augmented reality
Object detection
Unsupervised domain adaptation
You only look once

DOI: https://doi.org/10.30765/er.2684

Abstract:

Object detection plays a crucial role in enhancing mobile augmented reality (MAR) applications, but the computational limitations of mobile devices and the dynamic real-world environments pose significant challenges. Current literature often falls short in proposing solutions that maintain both high object detection accuracy and computational efficiency on mobile platforms. This study proposes a novel framework that integrates Knowledge Distillation (KD) and Unsupervised Domain Adaptation (UDA) to address these issues. KD transfers knowledge from a resource-heavy "teacher" model to a lightweight "student" model optimized for mobile deployment, while UDA enables the student model to adapt to real-world conditions without labeled data. Our framework uses YOLOv5 models, where the student model, YOLOv5n, learns from the teacher model, YOLOv5 small, improving precision and maintaining efficiency. Experiments on the VOC2007 and COCO datasets show that our SKD-UDA net achieves 78.2% mAP at IoU 0.5 and 50.8% mAP at IoU 0.5:0.95, outperforming the baseline YOLOv5n by 5.5% and 5.7%, respectively, without increasing the model size (1.9 MB). This approach enhances accuracy and computational efficiency, making it ideal for MAR applications. Our contributions advance object detection in MAR, improving user interaction by increasing detection accuracy, inference speed, and seamless integration of digital and physical environments.

1 Introduction

Augmented reality (AR) is a user-machine interaction technology that has recently gaining significant attention [1-4][1-4]. With the prominence of mobile phones and edge devices, mobile augmented reality (MAR) applications are increasing. Contrary to personal computer applications, a built-in camera capable of collecting images in real-time is included in MAR applications despite their lower memory and processing power compared to personal computers [5]. As a critical task in MAR, object detection has been a noteworthy computer vision research field [6]. The primary research objectives are to design an object detection framework that achieves high accuracy and computational efficiency while being suitable for deployment on mobile devices within MAR applications. We hypothesize that: (1) integrating Knowledge Distillation (KD) and Unsupervised Domain Adaptation (UDA) can enhance object detection accuracy on mobile platforms without increasing the model size, and (2) existing models can be adapted to dynamic MAR environments without additional labeled data. Despite advancements in deep learning, which have significantly improved object detection model accuracy [7-10], these models are frequently too large for practicality in MAR environments

[*] Corresponding author
E-mail address: esther@ukm.edu.my; hasankamrul@ieee.org

[11]. Edge devices with limited memory and computational resources must maintain high accuracy and low latency amidst changing conditions [12, 13]. Measurement is commonly carried out in the object detection process in MAR applications in two aspects: 1) efficiency indicates the speed of an object's recognition, which is primarily affected by the model size placed in the MAR application; 2) robustness reflects the accuracy when the model is used in different domains, such as rotation, scale, background, and lighting changes. Therefore, designing an object detection model with a small size and high accuracy is imperative.

The rapid development of object detection techniques has led to the proposal of many detection models. Most of these models yield high accuracy depending on the growing model parameters, such as Faster RCNN [14], Relation DETR [15], CAFF-DINO [16], and others. With the development and widespread application of transformer technology, more and more object detection models are using transformer technology to improve the accuracy of the model. However, the model parameters are increasing. For example, Co-DETR [17] with Swin-L as a backbone achieved 64.7 mAP on COCO dataset with 218 MByte parameters. FocalNets [18] combined with FocalNet-Huge backbone, Focal-Stable-DINO achieved 64.8 mAP on COCO with 689 MByte parameters. InternImage [19] achieved 65.4 mAP with 2180 MByte parameters. As shown above, every improvement in model accuracy depends on an increase in model parameters. Due to the large number of parameters in these models, they are difficult to deploy on mobile devices and, therefore cannot be used for MAR applications. Although many model compression techniques can be applied to these large models, they still cannot significantly reduce model parameters, which are too large for MAR applications. Knowledge distillation (KD) [20-25] is a model compression technique that is used to transfer knowledge from an extensive teacher network to a small network without risking performance [26, 27]. The KD is divided into response-based knowledge, feature-based knowledge, and relation-based knowledge. From larger teacher models to smaller student models, these knowledge types are transferred into logits outputs, middle feature layers, and diagrams, respectively [28, 29]. Precisely, response-based knowledge [30, 31] shifts the knowledge from the last logit outputs of the teacher model to the student model. However, the logits layer constantly increases the loss functions on the training stage, which are composed of cross-entropy loss of boxes classification, generalised intersection over union (GIOU) [32] loss of boxes regression, and distillation loss, specifically the Kullback-Leibler Divergence (KLDiv) of teacher and student's logits output. While response-based knowledge is widely used in relatively simple tasks and has achieved satisfactory outcomes, its use in object detection tasks would reduce the accuracy of the student model.

The knowledge of the feature-based model [33, 34] is transferred from the output of the intermediate layers or the last layer, which includes a high proportion of parameters. Following the intermediate layer accounted for a large proportion of the model parameters, feature-based knowledge could significantly reduce the model size or improve efficiency. However, given the unequal intermediate layers of the teacher and student models, selecting the hint layer from the teacher model to correctly coordinate the feature representations of the guided layer from the student model is challenging [35]. Without an appropriate selection, the student network would be excessively regularised, which decreases the accuracy of the student model. Response-based and feature-based knowledge use either the logits of the last layer or intermediate layers as knowledge transfer from the teacher to the student model. On the other hand, relation-based knowledge [36, 37] utilizes the outputs of any layers that compress the parameters of each object detection module. This article proposes a response-based KD method known as multitask loss fusion (MLF). The mean squared error (MSE) objective function calculates the regression and identification of object boxes, while the KLDiv estimates the object classification loss. To merge the three task losses, the sigmoid function will convert the object feature into an attention mask, followed by multiplying this mask through task loss. Object detection in MAR scenarios is a challenge due to the susceptibility of model accuracy to the environment, such as rotation, scale, background, and lighting change, among others. Due to the requirement of the new domain's adaptation and robustness in detection accuracy, unsupervised domain adaptation (UDA) has been widely used in the detection model to mitigate the gap between domains. Given that the source data is collected from different domains in real scenarios, multiple domains adaptation [20, 38, 39] with a unified training framework is proposed to solve the issue of domain shift. Subsequently, a generative and self-supervised domain adaptation method is suggested to manage the poor performance of different domain data [40]. Traditional UDA models require either the collection of images from the source domain and annotated images from the transferred target domain in advance or the acquisition of the source domain model and target domain model to learn from different domains. Rapidly changing and complex scenarios for real-time MAR applications make collecting labelled and unrecognized target domain images difficult. Moreover, trained MAR object detection models are generally not open source, and obtaining models in real-time application scenarios is challenging. Therefore, proposing a UDA method that does not rely on additional target domain labelled data and models in MAR applications is imperative.

The use of unsupervised image translators' technique to generate an unlabelled artificial dataset has shown significant improvements in domain adaptation. This is followed by using a dataset as a target domain dataset to train the target domain model [41]. However, the existing methods require additional datasets due to the challenges in collecting the target domain datasets in real industry scenarios. Labelling many images involves time and labour costs, with the synthetic data showing a different distribution than actual data. A target domain data-free UDA strategy known as target-data-free feature alignment (TFA), conducted in a teacher-student structure, is proposed to solve this issue. The teacher model features are regarded as the target domain, while the student model features are regarded as the source domain. Through this method, the TFA can maintain each source image feature distinction and enhance the adaptation of multiple domains. In building a high-speed and accurate object detection model in MAR applications. The subsequent sections of this paper are structured as follows: Section 2 outlines the motivation behind this study, emphasizing the challenges in object detection and the need for improved generalization across diverse datasets. Section 3 provides an extensive review of related work, covering significant advancements in object detection, including You Only Look Once (YOLO), KD, and UDA. Section 4 introduces the proposed framework for object detection, which enhances model generalization by training student models to recognize consistent features across varied datasets, leveraging a teacher network for knowledge transfer. Section 5 details the experimental setup, describing the standard datasets and methodologies used. Section 6 presents the results, offering a comprehensive comparison with existing frameworks. Section 7 delivers an in-depth discussion of the framework's key features, advantages, and limitations, along with recommendations for future research directions. Finally, Section 8 provides the conclusion, summarizing the key contributions and the framework's impact on advancing the field of object detection.

2 Motivation

Augmented reality (AR) is a user-machine interaction technology that has recently gaining significant attention [1][1-4]. With the prominence of mobile phones and edge devices, mobile augmented reality (MAR) applications are increasing. Contrary to personal computer applications, a built-in camera capable of collecting images in real-time is included in MAR applications despite their lower memory and processing power compared to personal computers [5]. As a critical task in MAR, object detection has been a noteworthy computer vision research field [6]. The measurement of the object detection process in MAR applications [1, 5] is commonly carried out in two aspects [42]: efficiency indicates the speed of the recognition object, primarily affected by the model size placed in the MAR application. On the other hand, robustness reflects the accuracy of the model's use in different domains, including rotation, scale, background, and lighting changes. With the transfer of the knowledge from the teacher model to the student model, the previous KD [20-24, 29, 33, 34, 36, 54-59, 66-68] methods focus on reducing the size of the student model without a sharp decline in the accuracy of the teacher model and inability of managing the domain shift issue. To illustrate this point, when the teacher model lacks different domain information, the student model would have no means of learning. Despite the suggestion of various UDA object detection methods [38-41, 61-64, 69-71] to solve domain shift problems, the target model remains vast and complex, which is unsuitable for MAR applications. Moreover, traditional UDA models typically require either annotated target source data from different domains or pre-trained target domain models, which may not be feasible for MAR applications. After surveying the advantages and limitations of KD and UDA, a combination of these techniques was proposed to design a MAR object detection model with a small size and high domain accuracy.

The main contributions of this paper include:

a) Integration of KD and UDA within the object detection pipeline, a novel application not widely explored before, which enables the suggestion for the training of a detection model with a small size and high generalization ability on new datasets;

b) A multi-tasking loss fusion (MLF) mechanism, which encourages the promotion of each task;

c) A teacher-student architecture for online learning that improves domain adaptability without the need for additional labeled data, a major limitation in existing approaches.

3 Related Work

3.1 You Only Look Once (YOLO)

You Only Look Once (YOLO) is a neural network-based algorithm that performs object detection in real-time. It was first proposed in 2015 and has been updated several times with versions such as YOLOv5 [43], YOLOv6 [44], and YOLOv7 [45], with YOLOv8 [46] being the most recent one. Unlike other object recognition algorithms that scan the input image multiple times [47, 48], YOLO achieves high speed by splitting the input image into a grid of cells and identifying the objects in each cell. It directly regresses the bounding box coordinates and class probabilities from the image pixels [46]. It also uses a single convolutional neural network to predict multiple bounding boxes. YOLO revolutionized the field by converting the object detection problem from classification to regression. YOLOv6, YOLOv7, and YOLOv8 obtain higher detection accuracy than YOLOv5, but their model size is largely increased. Using the Nano model as a reference, the sizes of YOLOv5, YOLOv6, YOLOv7, and YOLOv8 are 1.9 MB, 4.7 MB, 72.1 MB, and 3.2 MB, respectively. As the data shows above, the model size of all versions of YOLO is more significant than that of YOLOv5. This reduction in inference speed and increase in memory usage can make it difficult to deploy on mobile devices for MAR applications. YOLOv5 Nano (YOLOv5n) is the most miniature model in the YOLOv5, YOLOv6, YOLOv7, and YOLOv8 families, designed for edge and IoT devices [49-51]. It is less than 2 M bytes in model size and has a 45 millisecond CPU inference speed and a 45.7 mAP50 score on the COCO validation dataset. This makes it ideal for MAR applications, as it balances speed and accuracy.

The YOLOv5 object detection framework consists of a feature extract layer (backbone), a neck, a detection head layer, and a loss function. The network of CSPDarknet [52] with SPP neck and PANet [53] are employed as the backbone and detection head, respectively. The neck connects the backbone and the head. It comprises three convolution layers that predict the location of the bounding boxes (x, y, height, width), the scores, and the object classes. The YOLOv5 algorithm employs a feature extraction process to generate relevant data from input images, which is subsequently utilized by a prediction system to accurately identify and classify objects within the image, delineating their boundaries with bounding boxes. The YOLOv5 algorithm employs a sophisticated loss function that combines objects, class probability, and bounding box regression scores to identify and classify objects within an image accurately. Specifically, YOLOv5 has utilized Binary Cross-Entropy with logits loss function to calculate the loss for class probability and object score, while the location loss is determined using the Complete Intersection over Union (CIoU) loss. The object score measures the likelihood of an object being present within a bounding box. In contrast, the class probability score indicates the probability of an object belonging to a specific class. The bounding box regression score, on the other hand, reflects the accuracy of the predicted bounding box concerning the ground truth bounding box. The goal of the loss function is to minimize the discrepancy between the predicted and ground truth values for these three scores. This is accomplished by calculating the squared difference between the predicted and ground truth values for each score, summing them up to obtain a single value representing the total loss, and using this value to update the model's weights during forward propagation to enhance performance. During backpropagation, the gradients of the loss function to each weight in the network are calculated. These gradients are then used to update the network weights to minimize the loss function. This process is repeated for many training iterations until the model converges to a good solution. Figure 1 summarizes the YOLOv5 framework.

The blue arrow in Figure 1 represents the forward propagation, and the purple arrow represents the backpropagation process. The orange represents the total loss function, composed of CIoU loss [54] for bounding box regression, BCE loss for objectness score, and BCE loss for class probability. The purple arrows indicate that the parameters of each layer are updated by loss backpropagation. Furthermore, YOLOv5 comprises five different model configurations based on its model sizes, namely nano model (YOLOv5n), small model (YOLOv5s), medium model (YOLOv5m), large model (YOLOv5l), and extremely large model (YOLOv5x). Notably, YOLOv5 has been noticed for its higher performance compared to YOLOv4 in terms of precision and speed. It also achieves a state-of-the-art object detection algorithm [55, 56]. The loss function of YOLOv5 could be classified into three parts [68]: bounding box regression loss, object classification BCE loss, and object confidence loss.

The main variables involved in the YOLOv5 of the below formulas include:

i,j: These represent the coordinates of the bounding box, indicating the center of the object detected within the image. In YOLOv5, these are normalized to the dimensions of the image.

w,h: These variables represent the width and height of the bounding box, indicating the size of the object. These values are also normalized to the image dimensions.
p: This is the confidence score for each bounding box, representing the likelihood that the detected object corresponds to a particular class.
c: This variable represents the classification probability, which is the probability that the detected object belongs to a particular class.
Eq (1) presents the total loss function of YOLOv5. Eqs (2), (5), and (6) include the bounding box regression loss, object classification BCE loss, and object confidence loss, respectively. In this case, λ_{coord}, λ_{class}, λ_{noobj}, and λ_{obj} represent the coefficient of box regression, object classification, no object, and object, respectively that control the relative importance of each loss term. In the context of our proposed framework, these formulas play a crucial role. The bounding box regression loss ensures that the bounding boxes are accurately predicted in terms of both position and size, which is essential for high-precision object detection in MAR applications. The object classification BCE loss helps the model accurately identify objects, which is important in environments where multiple object classes need to be recognized reliably under varying conditions.Finally, The object confidence loss enhances the robustness of the detection by improving the model's ability to distinguish between object and non-object regions, crucial for real-time processing where false positives must be minimized. When the anchor box at the coordinate (i,j) contains ground-truth objects, the value $I_{i,j}^{obj}$ amounts to 1; otherwise, the value amounts to 0. In Eq (3), b and b^{gt} represent the prediction box and ground-truth box, respectively. In Eq (4), w^{gt}, h^{gt}, w, and h represent the ground-truth box width, ground-truth box height, prediction box width, and prediction box height, respectively. Based on Eq (5), $p_i(c)$ and $\hat{p}_l(c)$ represent the category probability of the prediction object and the ground-truth category, respectively. As for Eq (6), c_i and $\hat{c}_l$ denote the confidence score and the intersection of the prediction boundary box and ground-truth box, respectively.

$$Loss = L_{REG} + L_{CLS} + L_{OBJ} \tag{1}$$

$$L_{REG} = L_{CIoU} = 1 - IoU + \frac{p^2\left(b, b^{gt}\right)}{c^2} + av \tag{2}$$

$$IoU = \left|\frac{b \cap b^{gt}}{b \cup b^{gt}}\right| \tag{3}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{4}$$

$$L_{CLS} = \lambda_{class}\sum_{i=0}^{s^2}\sum_{j=0}^{B} I_{i,j}^{obj}\sum_{C\in class} p_i(c)\log\left(\hat{p}_l(c)\right) \tag{5}$$

$$L_{OBJ} = \lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B} I_{i,j}^{noobj}\left(c_i - \hat{c}_l\right)^2 + \lambda_{obj}\sum_{i=0}^{s^2}\sum_{j=0}^{B} I_{i,j}^{obj}\left(c_i - \hat{c}_l\right)^2 \tag{6}$$

In our framework, the formulas are not just used to optimize the YOLOv5 model but are combined with the knowledge distillation (KD) and unsupervised domain adaptation (UDA) strategies to further enhance the model's accuracy, robustness, and computational efficiency. The KD loss, as part of the distillation process, helps transfer knowledge from the teacher model to the student model, improving performance while maintaining a smaller model size. On the other hand, the UDA loss aids in adapting the model to new, unseen domains without the need for additional labeled target domain data, ensuring that the model remains robust across diverse MAR environments. Thus, the interaction between these variables and formulas within the YOLOv5 framework allows our proposed system to achieve the dual objectives of high accuracy and computational efficiency, making it suitable for real-time object detection on mobile devices.

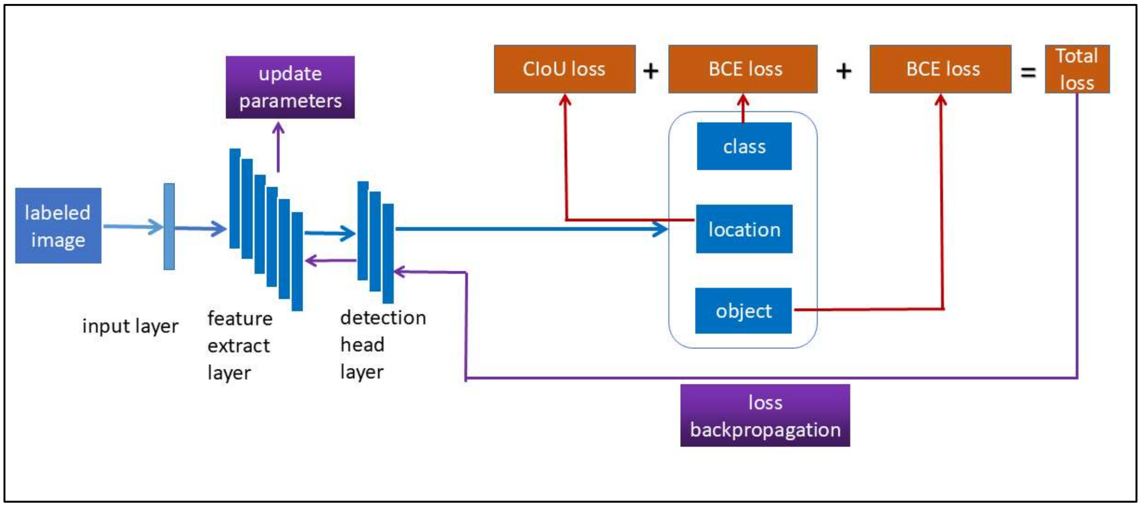


Figure 1. The YOLOv5 object detection framework. The object detection total losses comprise CIoU regression loss [51], object classification binary cross-entropy (BCE) loss, and object identity BCE loss.

3.2 Knowledge Distillation (KD) for Object Detection

KD based on transfer learning [57] has been used for the shift of knowledge from a large teacher network to a small student network for improvement in the performance and reduction of the size of the student model [20-24]. Therefore, the student network is designed with fewer parameters or a shallow layer. It is also trained by the labelled data and the larger model's knowledge output. This article proposes online teacher and student KD [58, 59] based on YOLOv5, the state-of-the-art detection algorithm. In line with this method, Guo et al. [24, 60] suggested using an ensemble of the soft targets of all student models to reduce the domain gap. This means that the outputs of multiple student models are combined to produce a more accurate prediction. Meanwhile, Kim et al. [61] employed a feature fusion learning (FFL) module to combine and generate meaningful feature maps of all the sub-networks online. This approach allows for integrating information from multiple sources to improve the model's accuracy. Additionally, Zhang et al. [62] introduced process-driven learning, which extends outside outcome-driven learning for augmented online KD based on adversarial mutual learning. This approach uses adversarial training to improve the performance of the model by forcing it to learn from its mistakes. Overall, these methods demonstrate how KD can improve the performance of object detection models in various ways without adding additional model parameters.

3.3 Unsupervised Domain Adaptation (UDA) for Object Detection

The UDA attempts to understand the domain-invariant representation to reduce the effect of domain shift and improve the robustness in the accuracy of the new domain [63]. The domain adaptive object detection aims to improve the generalization performance, which takes place in the prediction stage in many cases instead of the model train stage. Yu et al. [64] suggested a semi-supervised UDA learning method to learn better representations from cross-domains and reduce the content distribution gap. A weak self-training (WST) and adversarial background score regularisation (BSR) module is applied to reduce domain shift. Specifically, the WST diminishes the side effects of error pseudo labels, while BSP extracts the distinct features of domains [65]. This approach helps improve the model's performance by reducing the negative impact of incorrect pseudo labels and extracting features specific to the target domain. To overcome the negative transfer of features caused by adversarial training, VS et al. [66] employed global and category-aware domain adaptation simultaneously, allowing the attributes to be learned by the discriminator. This approach helps overcome the negative feature transfer caused by adversarial training. Subsequently, [67] addressed this issue by introducing conditional adversarial learning, in which the learning strength of well-aligned and poorly aligned samples are adjusted dynamically. This approach helps improve the model's performance by dynamically adjusting the learning strength of samples based on their alignment with the target domain. Overall, these methods demonstrate how UDA can improve the performance of object detection models in various ways.

4 Results and discussion

The Editor/Editorial Board may reserve the right to decide whether a paper is acceptable for publication, and if necessary, may require changes to the content, length, or language. This article suggests a unified object detection framework combined with KD and UDA. While the combination of these two techniques has been widely applied in image classification and segmentation tasks, achieving notable success [20, 70, 75-77], their application to object detection remains scarce. Object detection is a complex task, encompassing multiple stages such as the backbone, neck, box regression, and box classification networks, which makes it challenging to combine these components into a single, unified framework. To address this challenge, we propose an end-to-end teacher-student online learning architecture that integrates both KD and UDA in a coherent manner, optimizing the detection model for mobile deployment.

The loss function comprises KD loss, UDA loss, and object detection loss. Specifically, the KD loss includes the following: object coordinate regression, object classification, and object identity, which are subjected to Mean Squared Error (MSE) loss, Kullback-Leibler Divergence (KLDiv) loss, and MSE loss, respectively. These losses ensure the distillation of knowledge from the teacher model to the student model in terms of both object detection performance and feature presentation. Subsequently, the student model’s domain adaptability is enhanced through the Multiple Kernel Maximum Mean Discrepancy (MKMMD) loss [78] as part of the UDA loss. Although the MKMMD loss function adds complexity during training, it is not utilized during the inference phase. As a result, the inclusion of the MKMMD loss function during training does not increase the model's parameters at inference, ensuring that the inference speed remains unaffected. Unlike traditional methods, in our framework, the student model’s features are regarded as the source domain, while the teacher model’s features are regarded as the target domain [67]. This alignment forces the student model to adopt the new domain features extracted by the teacher model. In this way, the student model learns to generalize to new domains without requiring additional labeled target domain data, which is typically challenging to collect in real-world industrial environments. By combining KD and UDA, the proposed framework leverages the strengths of both techniques. KD helps the student model retain high accuracy despite a smaller model size, while UDA facilitates domain adaptation without the need for additional labeled data, making the framework robust in dynamic and resource-constrained environments like MAR applications. This unified approach significantly improves both model performance and generalization, addressing the key challenges of deploying efficient and accurate object detection models on mobile devices.

The key innovation in this approach lies in how the student and teacher models are aligned in terms of domain adaptation: Contrary to previous works, where the source domain and target domain are often interchanged or require additional labeled target domain data, in our framework, the student model features are regarded as the source domain, and the teacher model features are regarded as the target domain. This unique arrangement forces the student model to adopt new domain features extracted from the teacher model, eliminating the need for additional labeled target domain data. This is particularly valuable in industry environments where collecting labeled target domain data is both difficult and time-consuming. Overall, Figure 2 summaries the proposed framework of this study. The green, blue, and tangerine parts in Figure 2 are teacher model (YOLOv5s) [55, 56] student model (YOLOv5n) [55, 56] and loss function, respectively. The purple arrows indicate that the parameters of the student model are updated by total loss backpropagation, while the parameters of the teacher model are frozen during the training process. The total proposed framework loss function comprises bounding box location loss, object identity loss, and object classification loss. The bounding box location loss comprises MSE loss, MKMMD loss, and the original CIoU loss, representing the KD loss, UDA loss, and bounding box regression loss of the original YOLOv5n student model, respectively. The object identity loss comprises MSE loss, MKMMD loss, and BCE loss, representing the KD loss, UDA loss, and object identity loss of the original YOLOv5n student model, respectively. The object classification loss comprises KLDiv loss, MKMMD loss, and student BCE loss, representing the KD loss, UDA loss, and object classification loss of the original YOLOv5n student model, respectively. The integration of the KD and UDA components is achieved by coupling KD-specific losses (e.g., MSE, KLDiv) with UDA-specific losses (e.g., MKMMD), ensuring both knowledge transfer and domain adaptation during training. This design allows the student model to learn from the teacher model's outputs while adapting to the new domain. KD ensures that the student model closely mimics the output distributions of the teacher model (YOLOv5s). Losses like MSE and KLDiv guide the student to replicate the teacher's predictions, capturing both semantic and spatial information. The UDA using losses like MKMMD, minimizes the domain shift between the source domain (on which the teacher was trained) and the target domain (new dataset). This enhances the student's ability to generalize effectively across diverse, unseen datasets. By incorporating MKMMD loss in each of the three total loss components, the UDA ensures the adaptation of student model features across domains while KD ensures the student retains the high-performance traits of the teacher. This dual optimization allows the student model to maintain high accuracy even in new domains.

The student model (YOLOv5n) achieves high generalization ability on new datasets while maintaining a small computational footprint due to the following:

a) Efficient Model Architecture: YOLOv5n is designed to be lightweight, with fewer parameters compared to YOLOv5s, making it suitable for resource-constrained environments.

b) KD and UDA Integration: The joint optimization of KD and UDA ensures that the smaller student model effectively learns domain-invariant features while retaining high predictive accuracy.

c) Loss Decomposition: The carefully designed loss functions allow the student model to balance performance across location, identity, and classification tasks, ensuring robustness even when trained on limited or highly varied data.

This design allows YOLOv5n to outperform its size-class competitors, achieving a superior trade-off between model size and generalization performance. The integration of UDA ensures domain robustness, while KD ensures efficiency in learning from the larger teacher model.

4.1 Knowledge Distillation (KD)

The proposed response-based teacher-student KD schema distills the output logits of the teacher and student model. The teacher and student models are trained with YOLOv5s and YOLOv5n configuration, then the teacher model's parameters are frozen. The student model updates its parameters through the total loss backpropagation. Notably, the KD loss is an important component of a total loss, which consists of teacher and student model object boxes regression MSE loss, object classification KLDiv loss, and object identify MSE loss. Some similar works applied the MSE loss for regression or classification [27, 79] and the KLDiv loss for classification [43, 69, 71]. In the case of the MLF, the roles of the proposed MLF are to calculate and fuse KD loss. The MSE objective is to calculate the box's regression and object identification loss, while the KLDiv is employed to calculate object classification loss. Eq (7) shows that to fuse and balance these three tasks, each task needs to multiply a balance factor represented by b. Where, z^{t-obj} indicates teacher model object identify logits, z^{t} states teacher model regression features, z^{s} denotes student model regression features, o^{t} indicates teacher model object identify logits, o^{s} represents student model object identify logits, c^{t} denotes teacher model object classification logits, and c^{s} indicates student model object classification logits. Furthermore, the $mean$ represents the mean value of each sample and τ represents the temperature scaling hyper-parameter. Following that, α , β , and γ are hyper-parameters to balance each task weight. L_{REG}, L_{OBJ}, L_{CLS}, and L_{KD} represent object regression loss, identify loss, classification loss, and total KD loss, shown in Eqs (8)-(11) respectively. In MLF, the MSE and KLDiv loss functions are employed to effectively integrate KD loss function. To effectively combine these losses, each task-specific loss is weighted by a balancing factor, represented by b. This weighting ensures that no single task dominates the optimization process, allowing the model to learn all tasks concurrently. By introducing the balancing factor, the loss fusion mechanism ensures that all three tasks—regression, identification, and classification—are optimized without overemphasizing any single aspect. The factor is tuned to prevent imbalanced gradients, which could hinder convergence. In summary, the integration of MSE and KLDiv loss functions, along with appropriate balancing factors, enables the model to effectively learn from the teacher model while adapting to new domains, leading to improved performance and generalization.

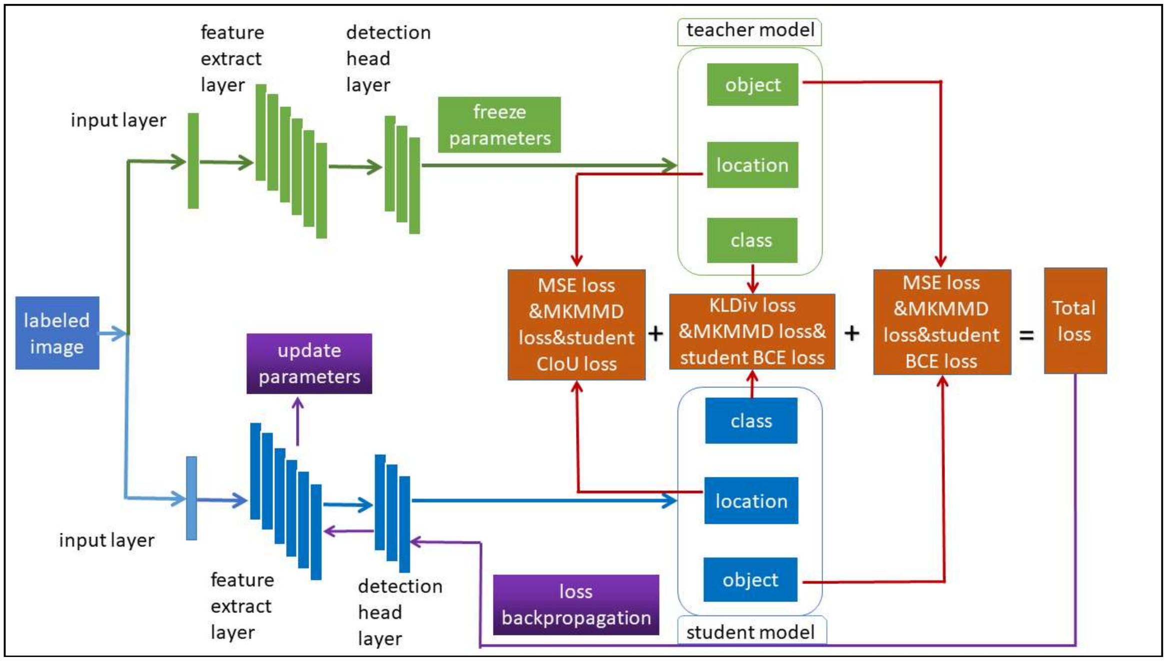


Figure 2. The proposed end-to-end online object detection framework. The KD loss comprises the MSE loss and KLDiv loss, while UDA loss includes the MKMMD loss [79], and object detection loss consists of the student model's CIoU regression loss [54], object classification BCE loss, and object identity BCE loss.

$$b = sigmoid\left(z^{t\_obj}\right) \tag{7}$$

$$L_{REG} = mean\left(\left|\left|z^{t} - z^{s}\right|\right|^{2} * b\right) \tag{8}$$

$$L_{OBJ} = mean\left(\left|\left|o^{t} - o^{s}\right|\right|^{2} * b\right) \tag{9}$$

$$L_{CLS} = mean\left(\tau^{2} * c^{t}\left(\tau\right) * \log\frac{c^{t}\left(\tau\right)}{c^{s}\left(\tau\right)} * b\right) \tag{10}$$

$$L_{KD} = \alpha * L_{REG} + \beta * L_{CLS} + \gamma * L_{OBJ} \tag{11}$$

4.2 Unsupervised Domain Adaptation (UDA)

The equation should be written in the MathType Equation Editor and positioned in the middle of the single text column. All equations should be numerated in brackets on the right side, as follows: Considering the real industry scenario, the collection and distinction between different domain data are not possible. However, with the model's capability of memorizing and enhancing each domain feature that appears in the train datasets, the generalisation ability in new or similar distribution datasets could be improved. Based on the article [67], it is a joint source domain data with target domain data to extract different level features, which do not require differentiation between source and target domains. Therefore, the student model features are regarded as low order and source level, while the teacher model's features are regarded as high order and target level. Besides, additional target domain datasets are not required. An MKMMD loss [78, 80] is present between the student and teacher model in implementing UDA loss. The MKMMD incorporates the learned feature representation into a reproducing kernel Hilbert space (RKHS) to enhance the ability of a single kernel employed by the maximum mean discrepancy (MMD) [80]. Furthermore, MMD [81-83] is the test statistic to determine whether the two distributions are the same and widely used in UDA. By identifying the function in the RKHS, the mean values of the two different distributions of the function are obtained. Subsequently, the mean dispersion is calculated by creating a difference between the two mean values with a source domain. Moreover, the Gaussian kernel function (RBF) is constantly used as the function. The two distributions are considered the same with an adequately small mean dispersion. Otherwise, different distributions would be created. The MMD loss is described as Eq (12).

$$MMD\left(D_s, D_t\right) = \frac{1}{N}\sum_{i=1}^{N} k\left(z_i^t\right) - \frac{1}{M}\sum_{j=1}^{M} k\left(z_j^s\right)_H \tag{12}$$

Where, D_s and D_t represent the source and target samples, respectively, while N and M indicate the number of the target and source domain samples, respectively. Furthermore, $k(.)$ denotes the kernel function, while z^t and z^s represent target and source features extracted by the neural network, respectively. Subsequently, H denotes the RKHS through the Gaussian kernel. The MKMMD [5-7] is incorporates multiple liner kernels to improve the representation ability of MMD. This article uses this function to improve the student model's new domain adaptability. Contrary to other articles that employ a minimum of two domain datasets, the teacher model logits D_t are considered the target domain, while the student model logits D_s are regarded as the source domain. The MKMMD loss function is described as Eq (13):

$$L_{MKMMD\left(D_s, D_t\right)} = E\left(f_K\left(D_s\right)\right) - E\left(f_K\left(D_t\right)\right)_H \tag{13}$$

Where, $f_k(.)$ is the sum of feature maps extracted by multiple kernels. Meanwhile, D_s and D_t represent the source and target detection logits at the end layer of the model, respectively as described as Eq (14).

$$K = \sum_{l=1}^{L} \lambda_u k_u \text{, s.t. } \lambda_u \geq 0 \text{, } \sum_{l=1}^{L} \lambda_u = 1 \text{, } u \in \{1, 2, ..., n\}. \tag{14}$$

Where L denotes the quantity of the kernel set, which is set to 3. The parameters λ and k, representing the weight and the magnitude of single gaussian kernel, were set to (1,0.5), (1,1), and (1,2), respectively.

4.3 Combination of KD and UDA

The total loss of SKD-UDA net is represented by L_T comprises three parts: object detection loss, KD loss, and UDA loss. Accordingly, the final loss of the proposed framework in this study is described as Eq (15).

$$L_T = \mu_1 {*} L_{RAW} + \mu_2 L_{KD} + \mu_3 L_{MKMMD} \tag{15}$$

Where, μ_1, μ_2, and μ_3 denote the trade-off parameters to balance the raw YOLOv5 loss, KD loss, and UDA loss. The student model was trained using L_{RAW}, the original YOLOv5 loss function, in addition to the KD loss depicted in Eq. 11 and the MKMMD loss illustrated in Eq. 13. The μ_1 controls the weight of the raw YOLOv5 loss, which includes bounding box regression, object classification, and object confidence scores. A higher μ_1 prioritizes standard detection tasks and ensures strong baseline performance on localization and detection. The μ_2 balances the influence of KD loss, which transfers knowledge from the teacher to the student model. This loss refines the student's predictions to match the teacher's distribution, improving performance on difficult-to-classify objects and enhancing overall accuracy. The μ_3 focuses on UDA by minimizing the domain shift between the source (teacher-trained) domain and the target domain. A higher μ_3 emphasizes the generalization capability of the model, particularly on unseen datasets. The trade-off parameters allow for adaptive optimization. A high μ_1 ensures that the model retains the foundational object detection capabilities of YOLOv5. A high μ_2 strikes a balance by enabling effective knowledge transfer without overshadowing domain adaptation. A low μ_3 provides sufficient domain adaptation without compromising detection and KD objectives. Fine-tuning these parameters is essential. For example, an excessively high μ_3 may result in over-adaptation to target domain features, thereby weakening detection performance on source-like data. Conversely, a low μ_3 could impair the model's ability to generalize effectively on new domains.

5 Results and discussion

5.1 Dataset

Extensive experiments were conducted to assess the proposed framework using the VOC2007 dataset [84] and Microsoft COCO dataset [85]. The training and validation sets were employed for the model training, while the testing set was employed for the model evaluation. Subsequently, the accuracy was measured by mAP0.5 and mAP0.5:0.95. Subsequently, the F1 score showed a positive correlation with mAP; in this case, a higher F1 score would be more favourable. The efficiency was evaluated by the model weights' size and the CPU and GPU speed. The ideal framework for MAR should present high mAP and F1 scores, a small model size, and low latency on CPU and GPU.

5.2 Baseline Comparison

This research compared the proposed SKD-UDA net with the original YOLOv5n [54], the YOLOv5n model with KD, and the YOLOv5s [54]. Among them, the Nano model serves as the baseline in terms of model size, model accuracy, CPU, and GPU inference speed. The results of the models are evaluated on the validation set of the VOC2007 and COCO datasets. Furthermore, both the YOLOv5n with KD and the YOLOv5n with KD and UDA shown in Table use a student-teacher framework, where the YOLOv5s is employed as the teacher model while the YOLOv5n is used as a student model. Based on the definition of response-based knowledge, a KD loss was incorporated at the end of the detector logits layer, defined in Eq (11). The proposed SKD-UDA net, which is the YOLOv5n with KD and UDA, shows similarity to the YOLOv5n with KD in terms of the structure of the student-teacher framework. The loss functions are conducted at the object's exact position, location, and classification layers, as shown in Fig. 2. However, while the KD loss is incorporated at the end of the detector logits layer of the YOLOv5n with KD as defined in equation (11), both MKMMD loss and KD loss are incorporated at the end of the detector logits layer of the YOLOv5n with KD and UDA, as defined in Eq (15).

5.3 Implementation

In this research, the state-of-the-art YOLOv5 was employed as a base detection model. In this case, the pre-trained checkpoint were loaded to initialize the model parameters, leveraging prior knowledge for enhanced performance. The training image size was set at 640 pixels, providing a balance between computational efficiency and detection accuracy, while mosaic augmentation, randomresizedcrop, flipping and domain randomization were applied to increase dataset diversity and improve robustness.

The initial learning rate was set to 0.0033 and decayed linearly over 100 epochs to ensure stable convergence while avoiding overfitting. The τ of Equation (10) was set at 2.0, while the α , β , and γ of Equation (11) were set at 1.0, 5.0, and 1.0, respectively which empirically to balance the contributions of the KD loss component. Specifically, τ was chosen to smooth the teacher model's probability distribution, aiding effective knowledge transfer, while β =5.0 was set higher to give more weight to critical loss components for aligning the teacher and student outputs. Following that, μ_1, μ_2, and μ_3 Equation (15) were set at 1.0, 1.0, and 0.5, respectively, representing the weights of the object detection module, KD module, and UDA module when the proposed framework was trained with object detection, KD, and UDA loss, determining to prioritize object detection and KD objectives while moderately incorporating UDA. Apart from that, the μ_1, μ_2, and μ_3 of Equation (15) were set at 1.0, 1.0, and 0, respectively, used to exclude the UDA component. The factors of the multitask loss fusion μ_1, μ_2, and μ_3 were determined through grid search, testing multiple configurations to identify a setup that minimized training loss without causing instability. These parameters were refined iteratively by observing validation performance and optimizing for generalization to unseen data. The selection process involved a combination of empirical testing and domain knowledge, aiming to balance the contributions of each loss term effectively and to ensure that the contributions of detection, KD, and UDA losses were aligned with the objectives of the proposed framework. The code was represented by Pytorch [75]. The experimental platform used in this study is shown in Table 1.

Table 1. Experimental Platform.

Name	Version
CPU	i9-14900HX
GPU	RTX2070Ti
Memory	32GB
Operating System	Windows 11
Deep Learning Framework	Pytorch1.8

6 Results

A comprehensive evaluation assessed the impact of KD and UDA loss, yielding several notable findings as summarized in Table 1 and Table 2. Particularly, as depicted in Eq 1, the YOLOv5n model was trained using the original YOLOv5 loss function, which includes bounding box regression loss, object classification BCE loss, and object confidence loss. The YOLOv5n model with KD, utilized a teacher-student architecture. In this setup, the KD loss comprises the MSE loss for object box regression, KLDiv loss for object classification, and MSE loss for object identification, as shown in Equation 11. The combination of KD and UDA (SKD-UDA), illustrated in Equation 15, was trained using object detection loss, KD loss, and UDA loss. The UDA loss corresponds to the MKMMD loss, as shown in Eq 13. The megabytes (M) were used to measure the size of the model, and the millisecond (ms) was used to measure the speed of the CPU and GPU of RTX2070Ti. This project involves deploying augmented reality applications on embedded devices, with the installation package on mobile devices not exceeding 100M. Object detection, a component of the application's functionality, has a model size constraint of 2M. Consequently, the YOLOv5n has been chosen for final deployment, as the model sizes of other options exceed 2M. Furthermore, the validation images were resized to 640 pixels. The ablation studies of each framework using VOC 2007 dataset and COCO dataset are shown in Table 2 and Table 3 respectively.

Table 2. Ablation Study of each framework Using VOC 2007 Dataset.

Training Methods	YOLOv5n	YOLOv5n with KD	YOLOv5n with UDA	SKD-UDA	YOLOv5s
Size (pixels)	640	640	640	640	640
mAP^{val} 0.5	72.8	76.6	74.6	78.2	84.6
mAP^{val} 0.5:0.95	45.1	48.1	46.9	50.8	58.5
Weights (M)	1.9	1.9	1.9	1.9	7.2
CPU Speed (ms)	46	46	46	46	99

Speed of RTX2070Ti	4.6	4.6	4.6	4.6	5.0

Table 3. Ablation Study of each framework Using COCO Dataset.

Training Methods	YOLOv5n	YOLOv5n with KD	YOLOv5n with UDA	SKD-UDA	YOLOv5s
Size (pixels)	640	640	640	640	640
mAP^{val} 0.5	45.7	48.2	47.1	49.6	56.8
mAP^{val} 0.5:0.95	28.0	30.7	29.9	32.8	37.4
Weights (M)	1.9	1.9	1.9	1.9	7.2
CPU Speed (ms)	46	46	46	46	99
Speed of RTX2070Ti	4.6	4.6	4.6	4.6	5.0

The ablation studies further highlight the contributions of each component, with the KD module boosting accuracy on challenging samples and the UDA module enabling domain generalization. Notably, the proposed SKD-UDA net outperformed the state-of-the-art YOLOv5n without introducing additional parameters to the source model, demonstrating its efficiency in leveraging existing architecture. Moreover, the SKD-UDA network surpasses the YOLOv5s in terms of accuracy on some samples from the validation dataset. Incorporating the KD loss into YOLOv5n resulted in a significant improvement of 3.8% in mAP0.5 and 3.0% in mAP0.5:0.95, reflecting the effectiveness of knowledge transfer from the teacher model. The inclusion of MKMMD-based UDA specifically ensures robust performance on datasets with significant domain shifts, where traditional models tend to falter. Additionally, the combination of KD and UDA in YOLOv5n led to a substantial increase in mAP0.5 from 72.8% to 78.2%, and mAP0.5:0.95 improved from 45.1% to 50.8%. Despite its lightweight design, the SKD-UDA net exhibited mAP performance comparable to YOLOv5s, while offering a smaller model size and faster execution speed on both CPU and GPU. Based on the results shown in Table 1, the YOLOv5n with the KD method could compress the size from YOLOv5s into YOLOv5n and at the same time can improve the accuracy. The SKD-UDA showed higher accuracy than the sole KD method without increasing the size of the Nano model. Additionally, the SDK-UDA showed a fast inference speed (efficient) that cost 46 ms in CPU mode and 4.6 ms in the GPU of RTX2070 Ti. This efficiency gain is critical for real-time applications and deployment on resource-constrained devices. The inference results of the four different models on VOC 2007 dataset are shown in Figure 3. Figure 3(a) shows the result of the YOLOv5n model, which only recognizes the object of sofa and fails to detect the two dogs. Figure 3(b) and 3(c) show the results of the YOLOv5n model with KD and the YOLOv5s model, respectively. Both can recognize the sofa but fail to detect one of the dogs, and they also misidentify another dog as a cat.

The results shown in Figure 3(d) for the YOLOv5n model with KD and UDA are encouraging. It indicates that the UDA module has effectively transferred knowledge from the teacher model, enhancing the model's ability to detect the object class "dog." This is a significant achievement, as it suggests that the model has improved its generalization capabilities, allowing it to recognize and locate "dog" objects more accurately, even in potentially new or varied data environments. Moreover, it shows that the proposed framework performs on par with the teacher model in some particular cases. The inference results of the four different models on COCO dataset are shown in Figure 4. The inference results presented are from the COCO dataset, which boasts a larger image collection than the VOC2007 dataset. Figure 4(a) demonstrates the performance of the original YOLOv5n model, which incorrectly identifies unrelated objects as balls. Figure 4(b) and 4(c) depict the YOLOv5n model enhanced with KD and UDA, respectively. These models outperform the standard Nano model by not misidentifying unrelated objects as balls. Moreover, as depicted in Figure 4(d), the proposed SKD-UDA net framework successfully corrects this error and enhances the object confidence score beyond that of the KD and UDA models, even when the object is small and blurry. This demonstrates the SKD-UDA module's exceptional capability in recognizing small and blurry objects. Its ability to outperform traditional KD and UDA models in terms of object confidence scores is quite impressive. In contrast, Figure 4(e), which shows the results from the YOLOv5 small model, erroneously identifies numerous irrelevant small blocks as people. The proposed framework demonstrates a lower false positive rate compared to the teacher model, thus reducing the likelihood of erroneous detections. Figure 5 displays the Precision-Recall curves for each class in the VOC2007 dataset, along with the average precision for each class.

The graph suggests that categories like ‘car,’ ‘horse,’ ‘bicycle,’ and others exhibit relatively good performance. In contrast, the ‘potted plant’ category underperforms compared to the rest. This could be attributed to the difficulty in extracting useful features from the ‘potted plant’ category, resulting in decreased accuracy. By addressing these challenges, we can aim to enhance the model’s precision and recall for the ‘potted plant’ category, thereby improving its overall accuracy. Figure 6 illustrates the evolution of several metrics during the training and validation phases, including box loss, object loss, and class loss. It also presents metrics such as accuracy, recall, mAP0.5, and mAP0.5:0.95 after each epoch. A steady decrease in loss metrics alongside improvements in accuracy, recall, and mAP scores is a strong indication that the model’s parameters are well-configured and that it’s learning effectively from the training data. Keep monitoring these metrics, as they will guide ours in making any necessary adjustments to the model or training process. Figure 7 present the confusion matrix on the VOC 2007 dataset that highlight some common challenges in object detection tasks, particularly in the context of a network like SKD-UDA. When objects such as ‘potted plant’, ‘chair’, and ‘bottle’ share similar features with the background, it can indeed lead to a higher Probability of Missed Detection (PMD). This is often due to these objects having less distinctive features or color patterns that blend with their surroundings. Similarly, the difficulty in distinguishing between ‘cow’ and ‘sheep’ could be attributed to their similar appearance in terms of shape, size, or coloration, which can confuse the network, leading to misclassification. In such cases, improving feature extraction methods and employing more sophisticated classification algorithms might help. Additionally, increasing the diversity and size of the training dataset to include more varied examples of these objects could potentially enhance the network’s ability to distinguish between them. By combining high detection accuracy with computational efficiency, the SKD-UDA net establishes itself as a practical and effective solution for real-world object detection tasks, outperforming state-of-the-art methods on key benchmarks. These results underline the framework’s ability to meet the dual objectives of precision and speed, setting a new standard for lightweight and adaptive detection models.

7 Discussion and Future Research Recommendations

In this research, the YOLOv5n model was employed as the student model due to the limitation of the memory size of edge devices in the MAR scenario. After using the proposed SKD-UDA net, the mAP0.5 of the source YOLOv5n model improved from 72.8% to 78.2%, although the parameters of the Nano model did not increase. The size of the proposed SKD-UDA net was 1.9 M, while the inference time was 46 ms on the CPU and 4.6 ms on the GPU. Given the model’s size and accuracy, using it in MAR applications with weak computation ability and small memory were a simple task. However, an accurate real-time response was required. Moreover, traditional response-based KD utilized the MSE loss function to calculate the logits of the teacher and student models, regardless of the varying tasks.

This led to the student model being unable to learn from the teacher model’s label completely. Theoretically, the KL divergence loss focuses on logit matching when τ increases and label matching when τ goes to 0 [86], the τ was defined in Eq (11). Given the complexity of object detection, which involves CIoU regression loss, object classification binary cross-entropy (BCE) loss, and object identity BCE loss, the MLF module in the proposed framework tailored logits to different tasks by using specific loss functions. In particular, the KD loss, UDA loss, and the original YOLOv5 loss function were integrated within the teacher-student architecture. For object classification, the KL divergence loss was chosen over the MSE loss, as KL divergence better captures differences between two probability distributions, making it more suitable for probabilities. The MSE loss, by squaring errors, produces very small changes in probability-based tasks, hindering learning. Additionally, the KL divergence loss includes a temperature factor (τ), which balances logit matching and label matching. In experiments, the smallest loss value was achieved when τ was set to 2, preventing overfitting. The MSE loss was used in the MLF module for CIoU regression and object identity tasks, as it is more sensitive to object coordinates. Furthermore, the object identification probability served as a multiplier to weigh the regression and classification task losses, significantly influencing the total loss function.

Thus, the MSE loss was chosen for the object identity task due to its sensitivity and ability to help the student model directly learn the teacher model’s logits. Contrary to the other UDA methods, the proposed

SKD-UDA net did not require the target domain dataset, which was challenging to collect and distinguish from the source domain where the teacher model is treated as target domain and student model is treated as the source domain. Instead, the teacher model is treated as the target domain, and the student model is treated as the source domain. The MKMMD is used as UDA loss that combines multiple kernels to measure the difference between teacher and student models probability distributions. The MKMMD is better than the MMD because it can use multiple kernels to capture more information about the distributions and make them closer. The MMD measures the difference between two probability distributions based on their average representations in a special space. But the MMD may not be able to handle the changes of class probabilities, the shapes of the object, or the different modes of the data. The MKMMD can deal with these problems by combining different kernels with different weights, which can match the distributions in different ways, addressing challenges such as class probability variations, object shape differences, and different domain data distributions. The MKMMD can also use a method to find the best weights for each kernel. Compared to a single YOLOv5n model, the YOLOv5n model with KD surpassed the mAP0.5:0.95 by 3%. However, the MLF module which is the combination of KD and UDA loss created a better result than the sole use of KD loss where the mAP0.5:0.95 score improved by 2.7%, indicating that the MLF module was practical. With the large memory size of the edge devices, it is suggested for the SKD-UDA net in future works to employ the YOLOv5 large model as the teacher to improve the accuracy. Similarly, the small model could be employed as the student for the same purpose. The proposed SKD-UDA net could also be applied to other object detection frameworks, such as YOLOv4 , YOLOv6, YOLOv7 and YOLOv8 to improve the source model's accuracy and inference speed. Therefore, designing a more accurate KD scheme and UDA loss function is important in the future work. For example, one could first use a small model for distillation, then use a larger model for distillation, and explore other UDA loss functions to improve the performance of our framework.

8 Limitations And Future Work

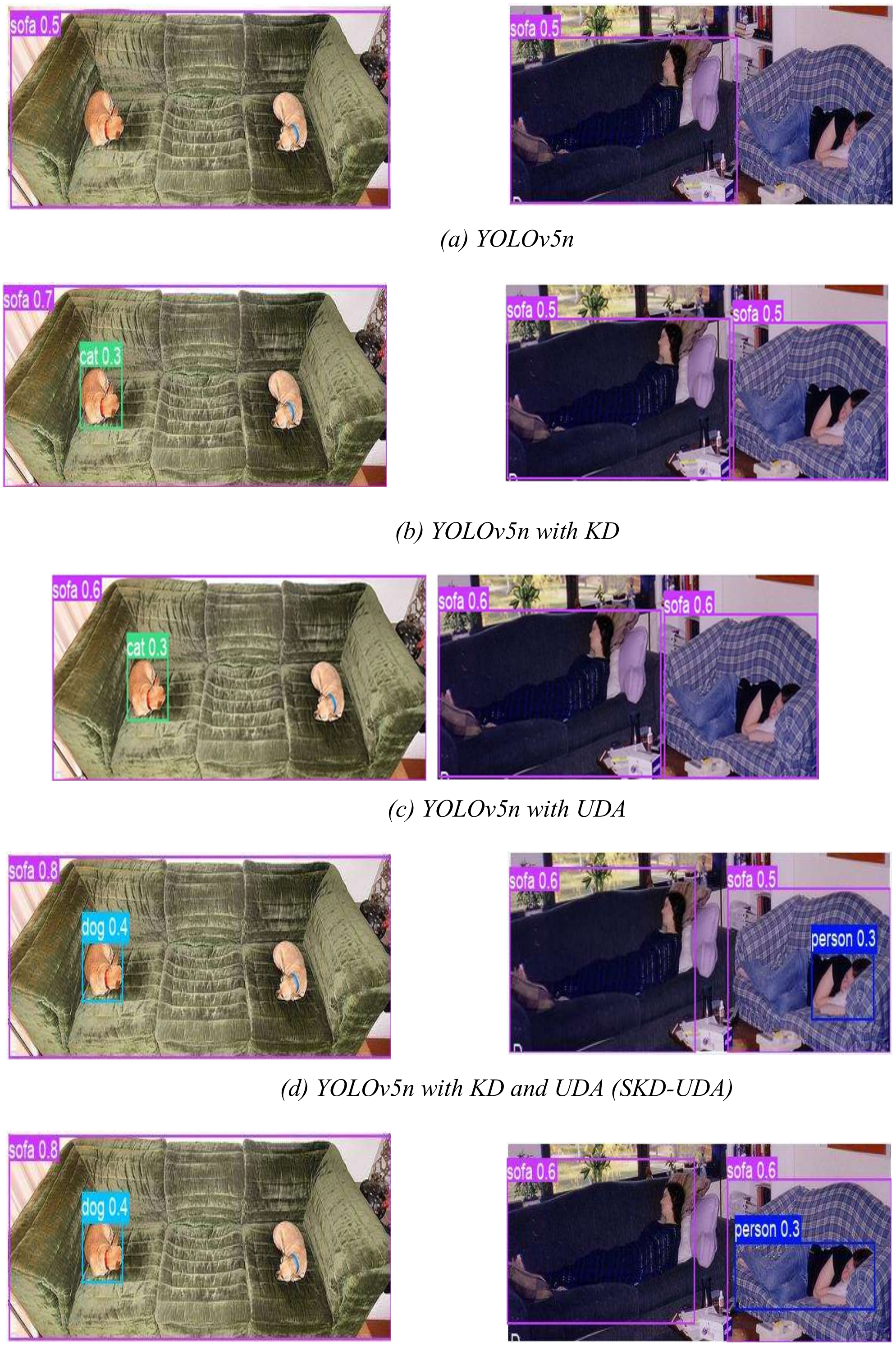
While the SKD-UDA net delivers significant improvements in accuracy and efficiency, its reliance on the YOLOv5n as the student model may limit its performance on more complex tasks. For edge devices with larger memory capacities, future research could explore employing the YOLOv5 large model as the teacher model to further enhance accuracy while using small models as students to maintain efficiency. Additionally, the SKD-UDA net could be extended to other object detection frameworks, such as YOLOv4, YOLOv6, YOLOv7, and YOLOv8, to improve both accuracy and inference speed across diverse applications.Future work should also focus on designing a more accurate KD scheme and UDA loss function. For instance, an approach involving iterative distillation, starting with a small model and progressively using larger models, could yield better results. Furthermore, exploring alternative UDA loss functions to capture domain discrepancies more effectively would enhance the adaptability of the framework. These directions present promising opportunities to expand the capabilities of the SKD-UDA net.

9 Conclusion

In this article, SKD-UDA net object recognition framework based on YOLOv5 was proposed to boost object detection in MAR. The proposed framework improves the robustness in precision and decreases the final model's size in MAR scenario for object detection. The SKD-UDA net, which has a 1.9M model size and is smaller than YOLOv6, YOLOv7, YOLOv8, and models based on transformers, was selected as the final model can work on edge devices with its very small size, fast speed, and good accuracy. Despite the small size of the final model, it exhibited domain adaptation ability and robust feature representation ability gained by the teacher model under the proposed framework. Notably, the proposed SKD-UDA net was suitable for MAR as it met the requirements of MAR that included high accuracy and efficiency. The proposed SKD-UDA net also presented important ideas for real-time object detection by improving precision and productivity in complex applications, such as automated driving cars, object tracking, and face detection.

Acknowledgement

The author acknowledges the Fundamental Research Grant Scheme (FRGS), grant number [FRGS/1/2022/ICT10/UKM/02/2], funded by the Ministry of Higher Education (MOHE), Malaysia and grant number [UKM-TR2022-03], funded by the National University of Malaysia.



(a) YOLOv5n

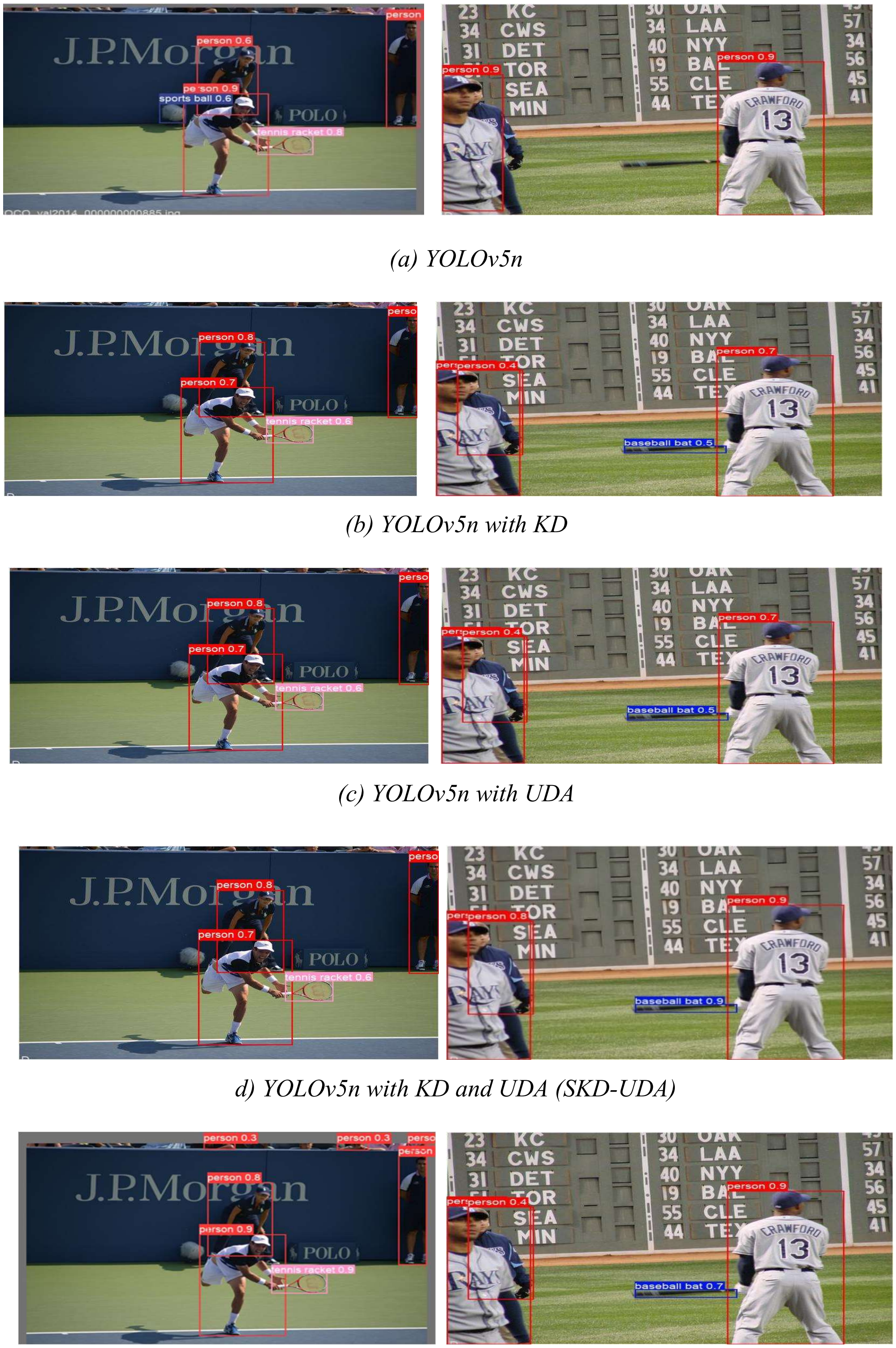
(b) YOLOv5n with KD

(c) YOLOv5n with UDA

(d) YOLOv5n with KD and UDA (SKD-UDA)

(e) YOLOv5s

Figure 3. Inference results of the four different models on VOC 2007 dataset. (a) Inference results of YOLOv5n; (b) Inference results of YOLOv5n with KD; (c) Inference results of YOLOv5n with UDA; (d) Inference results of YOLOv5n with SKD-UDA; (e) Inference results of YOLOv5s model.



(a) YOLOv5n

(b) YOLOv5n with KD

(c) YOLOv5n with UDA

d) YOLOv5n with KD and UDA (SKD-UDA)

(e) YOLOv5s

Figure 4. Inference results of the four different models on COCO dataset. (a) Inference results of YOLOv5n; (b) Inference results of YOLOv5n with KD; (c) Inference results of YOLOv5n with UDA; (d) Inference results of YOLOv5n with SKD-UDA; (e) Inference results of YOLOv5s model.

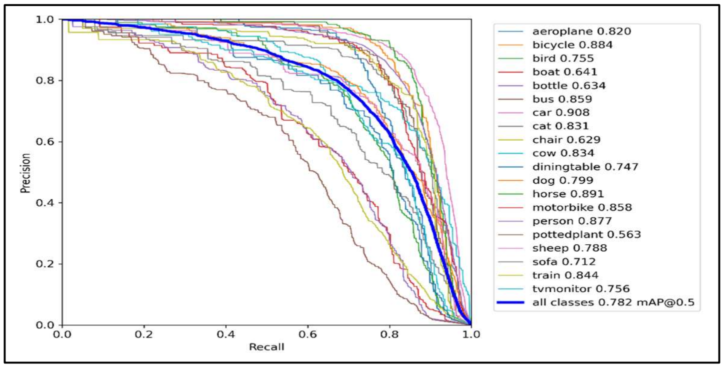


Figure 5. The Precision-Recall curve during training on the VOC 2007 dataset of the SKD-UDA network.

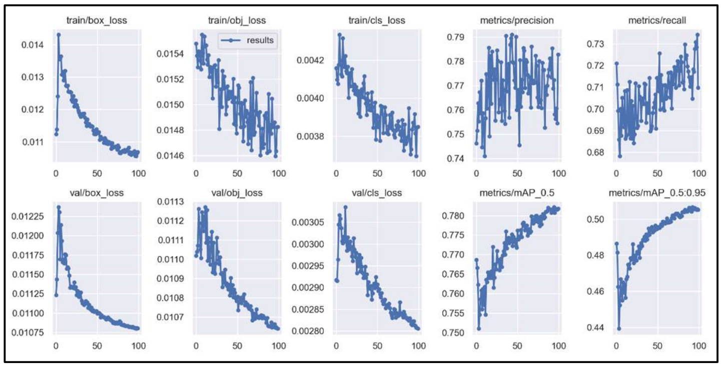


Figure 6. The SKD-UDA network convergence on the VOC 2007 dataset.

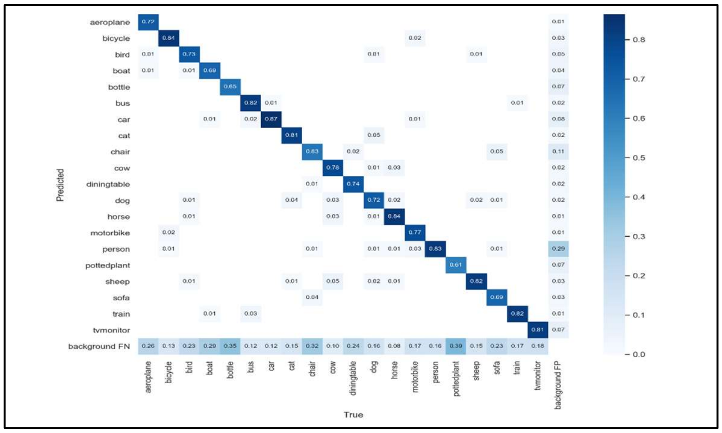


Figure 7. The confusion matrix on the VOC 2007 dataset[8].

References

[1] M. C. Lam, N. A. Suwadi, A. H. Mohd Zainul Arifien, B. K. Poh, N. S. Safii, and J. E. Wong, "An evaluation of a virtual atlas of portion sizes (VAPS) mobile augmented reality for portion size estimation," *Virtual Reality,* vol. 25, no. 3, pp. 695-707, 2021.

[2] Z. Mahayuddin and A. Saif, "Efficient hand gesture recognition using modified extrusion method based

on augmented reality," TEST Eng. Manag., vol. 83, pp. 4020–4027, 2020.

[3] Z. R. Mahayuddin, and A. Saif, "Vision based 3D Gesture Tracking using Augmented Reality and Virtual Reality for Improved Learning Applications," International Journal of Advanced Computer Science Applications, 2021.

[4] A. S. Saif, Z. R. Mahayuddin, and A. Shapi'i, "Augmented Reality based Adaptive and Collaborative Learning Methods for Improved Primary Education Towards Fourth Industrial Revolution (IR 4.0)," *International Journal of Advanced Computer Science Applications,* vol. 12, no. 6, 2021.

[5] S. Y. Tan, H. Arshad, and A. Abdullah, "An improved colour binary descriptor algorithm for mobile augmented reality," *Virtual Reality,* vol. 25, no. 4, pp. 1193-1219, 2021.

[6] X. Zeng, S. Y. Tan, and M. F. Nasrudin, "Adapt-Net: A unified object detection framework for mobile augmented reality," *IEEE Access,* vol. 12, pp. 120788–120803, 2024.

[7] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: a systematic review," *IEEE Access,* vol. 9, pp. 101574–101599, 2021.

[8] A. Baig, "Deep attributes and decisions fusion for no-reference video quality analysis," *Big Data Comput.* Vis., vol. 3, no. 3, pp. 91–103, 2023.

[9] A. Mageed, A. H. Bhat, and S. A. Edalatpanah, "Shallow Learning vs. Deep Learning: A Practical Guide for Machine Learning Solutions, " *Cham, Switzerland: Springer Nature,* 2024, pp. 77–91.

[10] Z. Khodaverdian, H. Sadr, S. A. Edalatpanah, S. Nasirzadeh, A. Shojaeinasab, and N. I. Ud Din, "An energy aware resource allocation based on combination of CNN and GRU for virtual machine selection," *Multimedia Tools Appl.,* vol. 83, no. 9, pp. 25769–25796, 2024.

[11] S. S. Hosseini, M. Yamaghani, and S. P. Arabani, "A Review of Methods for Detecting Multimodal Emotions in Sound, Image and Text," *Journal of Applied Research on Industrial Engineering*, 2024.

[12] J. Qu, Z. Gao, T. Zhang, Y. Lu, H. Tang, and H. Qiao, "Spiking Neural Network for Ultra-low-latency and High-accurate Object Detection," arXiv preprint arXiv:230612010, 2023.

[13] J. Seo, S. Jang, J. Cha, H. Choi, D. Kim, and S. Kim, "MDED-Framework: A Distributed Microservice Deep-Learning Framework for Object Detection in Edge Computing," *Sensors*, vol. 23, no. 10, p. 4712, 2023.

[14] R. Girshick, ed., *Proceedings of the IEEE international conference on computer vision*, 2015.

[15] T-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, eds., *Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision*, 2017.

[16] X. Hou, M. Liu, S. Zhang, Y. Li, B. Zhang, and J. Tang, "Relation DETR: Exploring Explicit Position Relation Prior for Object Detection," 2024.

[17] K. Helvig, B. Abeloos, and P. Trouve-Peloux, "CAFF-DINO: Multi-spectral object detection transformers with cross-attention features fusion," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 0–0. Accessed: Dec. 1, 2024. DOI: 10.1109/CVPRW63382.2024.00309.

[18] T. Ren, J. Yang, S. Liu, A. Zeng, F. Li, H. Zhang, and L. Wang, "A Strong and Reproducible Object Detector with Only Public Datasets," arXiv preprint arXiv:230413027, 2023.

[19] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, and L. Zhang, eds., *Internimage: Exploring large-scale vision foundation models with deformable convolutions. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[20] A. Belal, M. Kiran, J. Dolz, L-A. Blais-Morin, and E. Granger, "Knowledge distillation methods for efficient unsupervised adaptation across multiple domains," *Image Vision Computing*, vol. 108, p. 104096, 2021.

[21] W. Park, D. Kim, Y. Lu, and M. Cho, eds., *Relational knowledge distillation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[22] K. Kim, B. Ji, D. Yoon, and S. Hwang, eds., *Self-knowledge distillation with progressive refinement of targets. Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[23] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, eds., *Improved knowledge distillation via teacher assistant. Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[24] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Torr, eds., *Online knowledge distillation via collaborative learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[25] X. Dai, Z. Jiang, Z. Wu, Y. Bao, Z. Wang, S. Liu, and L. Lin, eds., *General instance distillation for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[26] L. Sun, J. Gou, B. Yu, L. Du, and D. Tao, "Collaborative teacher-student learning via multiple knowledge transfer," arXiv preprint arXiv:08471, 2021.

[27] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network (2015)," arXiv preprint arXiv:02531, 2015, 2.

[28] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[29] L. Chen, Y. Chen, J. Xi, and X. Le, "Knowledge from the original network: restore a better pruned network with knowledge distillation," *Complex Intelligent Systems*, pp. 1–10, 2021.

[30] V. Vats and D. Crandall, "Controlling the Quality of Distillation in Response-Based Network Compression," arXiv preprint arXiv:10047, 2021.

[31] T. Feng and M. Wang, "Response-based Distillation for Incremental Object Detection," arXiv preprint arXiv:13471, 2021.

[32] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, eds., *Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[33] R. Liu, K. Yang, H. Liu, J. Zhang, K. Peng, and R. Stiefelhagen, "Transformer-based Knowledge Distillation for Efficient Semantic Segmentation of Road-driving Scenes," arXiv preprint arXiv:13393, 2022.

[34] Y. Zhu and Y. Wang, eds., *Student customized knowledge distillation: Bridging the gap between student and teacher. Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[35] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," arXiv preprint arXiv:02531, 2014.

[36] H. Cheng, L. Yang, and Z. Liu, eds., "Relation-Based Knowledge Distillation for Anomaly Detection," *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 2021: Springer.

[37] J. Liu, H. Qin, Y. Wu, J. Guo, D. Liang, and K. Xu, "CoupleFace: Relation Matters for Face Recognition Distillation," arXiv preprint arXiv:05502, 2022.

[38] D. Zhang, M. Ye, Y. Liu, L. Xiong, and L. Zhou, "Multi-source unsupervised domain adaptation for object detection," *Information Fusion*, vol. 78, pp. 138–148, 2022.

[39] X. Wei, S. Liu, Y. Xiang, Z. Duan, C. Zhao, and Y. Lu, "Incremental learning based multi-domain adaptation for object detection," *Knowledge-Based Systems*, vol. 210, p. 106420, 2020.

[40] K. Fujii and K. Kawamoto, "Generative and self-supervised domain adaptation for one-stage object detection," *Array*, vol. 11, p. 100071, 2021.

[41] V. F. Arruda, R. F. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, N. Sebe, and A. Verri, "Cross-domain object detection using unsupervised image translation," *Expert Systems with Applications*, vol. 192, p. 116334, 2022.

[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, eds., *You only look once: Unified, real-time object detection—proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[43] N. Bhalotia, M. Kumar, A. Alameen, S. Venkateshan, and S. Sinha, "A helping hand to the elderly: securing their freedom through the HAIE framework," *Applied Sciences*, vol. 13, no. 11, p. 6797, 2023.

[44] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, and W. Liu, "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:220902976, 2022.

[45] C-Y. Wang, A. Bochkovskiy, and H-Y.M. Liao, eds., *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[46] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond," arXiv preprint arXiv:230400501, 2023.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015, 28.

[48] A. K. Nsaif, S. H. M. Ali, K. N. Jassim, A. K. Nseaf, R. Sulaiman, A. Al-Qaraghuli, and A. Hashim, “FRCNN-GNB: Cascade faster R-CNN with Gabor filters and naïve bayes for enhanced eye detection,” 2021, 9, pp. 15708–19.

[49] A. Bochkovskiy, C-Y. Wang, and H-Y.M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:10934, 2020.

[50] N. Osintsev and V. Nozick, “ITS-based wireless traffic monitoring solution,” *Big Data and Computing Visions*, vol. 3, no. 4, pp. 154–159, 2023.

[51] J. Pourqasem, D. Tešić, and E. Abdolmaleki, “Leveraging IoT and Industry 4.0 for Enhanced Environmental Safety,” *Computational Algorithms and Numerical Dimensions*, vol. 2, no. 4, pp. 234–239, 2023.

[52] W. Tun, J. Pourqasem, and S. A. Edalatpanah, “Optimizing resource discovery technique in the P2P grid systems,” *Wireless Communications and Mobile Computing*, vol. 2020, no. 1, p. 1069824, 2020.

[53] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, eds., “Panet: Few-shot image semantic segmentation with prototype alignment. Proceedings of the IEEE/CVF International Conference on Computer Vision,” 2019.

[54] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, “Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation,” IEEE Trans. Cybern., vol. PP, Epub Aug. 27, 2021. doi: 10.1109/TCYB.2021.3095305. PubMed PMID: 34437079.

[55] D. Thuan, “Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm,” 2021.

[56] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, eds., “TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. Proceedings of the IEEE/CVF International Conference on Computer Vision,” 2021.

[57] B. Shao and Y. Chen, “Multi-granularity for knowledge distillation,” *Image Vision Computing*, vol. 115, p. 104286, 2021.

[58] C. Tan and J. Liu, “Online knowledge distillation with elastic peer,” *Information Sciences*, vol. 2022, 583, pp. 1–13.

[59] M. Tzelepi, N. Passalis, and A. Tefas, “Online Subclass Knowledge Distillation,” *Expert Systems with Applications*, vol. 181, p. 115132, 2021.

[60] D. Walawalkar, Z. Shen, and M. Savvides, eds., *Online ensemble model compression using knowledge distillation. European Conference on Computer Vision*, 2020: Springer.

[61] J. Kim, M. Hyun, I. Chung, and N. Kwak, eds., “Feature fusion for online mutual knowledge distillation,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021: IEEE.

[62] X. Zhang, S. Lu, H. Gong, Z. Luo, and M. Liu, eds., “Amln: adversarial-based mutual learning network for online knowledge distillation. European Conference on Computer Vision,” 2020: Springer.

[63] G. Wei, C. Lan, W. Zeng, and Z. Chen, eds., “Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,” 2021.

[64] F. Yu, D. Wang, Y. Chen, N. Karianakis, T. Shen, P. Yu, and K. Wong, “Unsupervised domain adaptation for object detection via cross-domain semi-supervised learning,” arXiv preprint arXiv:07158, 2019.

[65] S. Kim, J. Choi, T. Kim, and C. Kim, eds., “Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision,” 2019.

[66] V. S. V, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, eds., “Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,” 2021.

[67] D. Guan, J. Huang, A. Xiao, S. Lu, and Y. Cao, “Uncertainty-aware unsupervised domain adaptation in object detection,” *IEEE Transactions on Multimedia*, 2021.

[68] J. Zhou, P. Jiang, A. Zou, X. Chen, and W. Hu, “Ship Target Detection Algorithm Based on Improved YOLOv5,” *Journal of Marine Science Engineering*, vol. 9, no. 8, p. 908, 2021.

[69] Y. Choi, J. Choi, M. El-Khamy, and J. Lee, eds., “Data-free network quantization with adversarial knowledge distillation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops,” 2020.

[70] L. T. Nguyen-Meidine, A. Belal, M. Kiran, J. Dolz, L-A. Blais-Morin, and E. Granger, eds., "Unsupervised multi-target domain adaptation through knowledge distillation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision," 2021.

[71] G. Xu, Z. Liu, X. Li, and C. C. Loy, eds., "Knowledge distillation meets self-supervision. European Conference on Computer Vision," 2020: Springer.

[72] Z. Li, L. Zhao, W. Chen, S. Yang, D. Xie, and S. Pu, eds., "Target-aware auto-augmentation for unsupervised domain adaptive object detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022: IEEE.

[73] P. Yuan, W. Chen, S. Yang, Y. Xuan, D. Xie, Y. Zhuang, and Q. Huang, eds., "Simulation-and-mining: towards accurate source-free unsupervised domain adaptive object detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022: IEEE.

[74] Y. Xu, H. Fan, H. Pan, L. Wu, and Y. Tang, eds., "Unsupervised Domain Adaptive Object Detection Based on Frequency Domain Adjustment and Pixel-Level Feature Fusion," in *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2022: IEEE.

[75] C-H. Chao, B-W. Cheng, and C-Y. Lee, eds., "Rethinking Ensemble-Distillation for Semantic Segmentation Based Unsupervised Domain Adaption. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition," 2021.

[76] E. Granger, M. Kiran, J. Dolz, and L-A. Blais-Morin, eds., "Joint progressive knowledge distillation and unsupervised domain adaptation," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020: IEEE.

[77] H. Cui, C. Yuwen, L. Jiang, Y. Xia, and Y. Zhang, "Bidirectional cross-modality unsupervised domain adaptation using generative adversarial networks for cardiac image segmentation," *Computers in Biology and Medicine*, vol. 136, p. 104726, 2021.

[78] M. Long, Y. Cao, J. Wang, and M. Jordan, eds., "Learning transferable features with deep adaptation networks," *International conference on machine learning*, 2015: PMLR.

[79] T. Kim, J. Oh, N. Kim, S. Cho, and S-Y. Yun, "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation," arXiv preprint arXiv:08919, 2021.

[80] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. Schölkopf, "Optimal kernel choice for large-scale two-sample tests," *Advances in neural information processing systems*, 2012, 25.

[81] J. Gu and V. Tresp, "Search for better students to learn distilled knowledge," arXiv preprint arXiv:11612, 2020.

[82] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, H. P. Schölkopf, B., and A. J. Smola, "Integrating structured biological data by Kernel Maximum Mean Discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–57, Epub 2006/07/29. doi: 10.1093/bioinformatics/btl242. PubMed PMID: 16873512.

[83] H. Chen, C. Wu, B. Du, and L. Zhang, "Dsdanet: Deep siamese domain adaptation convolutional neural network for cross-domain change detection," arXiv preprint arXiv:09225, 2020.

[84] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," 2007.

[85] T-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, eds., "Microsoft coco: Common objects in context. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13," 2014: Springer.

[86] T. Kim, J. Oh, N. Kim, S. Cho, and S-Y. Yun, "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation," arXiv preprint arXiv:210508919, 2021.