INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 1, PP. 1–10
Manuscript received July 18, 2025; revised February, 2026. doi: 10.24425/ijet.2026.157893

Efficient weapon detection using convolutional and transformer-based Deep Learning Models

Kamil Gomulka

***Abstract*—Detecting weapons in public spaces remains a significant challenge in computer vision and public safety applications. While deep learning models have achieved great progress in general object detection, there is still a lack of focused studies on class-specific detection tasks, in particular those using new architectures such as transformers. In this work, a comprehensive evaluation of the state-of-the-art deep learning object detection approaches is conducted, including convolution and transformer-based architectures. Therefore, a dedicated large-scale dataset that combines images from multiple public sources is introduced, with a focus on three main weapons categories, enabling a more targeted evaluation. Furthermore, in the paper, the effectiveness of the best-performing architecture is further improved with proposed modifications, including architectural changes and determining a suitable loss function. Finally, the obtained detection approach achieves superior detection results, as evidenced by all performance criteria.**

***Keywords*—weapon detection; object detection; deep learning; transformers; convolutional neural networks**

I. Introduction

With an increase in the availability of surveillance systems and emerging public safety concerns, there is a need for automated and reliable threat detection. Among potential security threats, the presence of weapons, including knives, firearms, and explosives, creates an increased risk in public places. Manual monitoring can be labor-intensive and result in human error, highlighting the need for reliable automatic detection systems. Therefore, to ensure their effectiveness, computer vision-based object detection has become a key component. In recent years, the rapid development of deep learning models has greatly improved the capabilities of such systems in a variety of scenarios, including the detection of handguns and knives in video surveillance images [1]. However, challenges remain in achieving high accuracy in diverse environments and lighting conditions, as well as dealing with difficult object appearances, especially in cases involving uncommon or partially occluded weapon types. It highlights the need for continuous refinement of existing models and exploration of more effective deep learning architectures.

Despite significant progress, current weapon detection approaches still face several key challenges. Most existing object detection models are trained on general-purpose datasets such as COCO [2] or Open Images [3]. These datasets include 80 classes, yet none are dedicated to specific weapon types. As a result, models must be fine-tuned to effectively identify such objects. Achieving high accuracy in challenging scenarios remains difficult, often leading to unreliable detection results. In addition, small everyday objects held by people can cause false positives. Since the quality of training data directly affects model performance, there is a growing demand for custom datasets specifically designed for weapon detection. This paper addresses this by preparing a combined dataset that includes images of credit cards, wallets, and purses, which allowed proper model training. Another important aspect is that, although several studies have explored weapon detection, the newer transformer-based model RT-DETRv2 has only been examined in one study, which focused solely on handguns. In contrast, this work extends the detection task to include three categories: gun, knife, and grenade. Moreover, while recent models have improved detection performance, there is still limited analysis of how architectural choices—such as alternative backbones beyond those originally proposed—impact the results. In this paper, two additional backbones are evaluated and compared with those used in the original implementation. Finally, specific loss functions are rarely examined in the context of weapon detection. The experiments carried out in this study offer a more comprehensive understanding of how both architectural and training strategies affect overall performance.

To address the mentioned gaps, in this paper, the detection of three classes of weapons is examined using convolutional and transformer-based models. The main contributions of this paper are:

- a combined dataset with unified annotations for three weapon categories (knife, gun, and grenade);
- adapting existing object detection models to detect specified classes not supported by pre-trained versions;
- comparison of multiple deep network architectures;
- successful modification of the best-performing RT-DETRv2 architecture to further improve the obtained detection result.

Kamil Gomulka is with Doctoral School of the Rzeszow University of Technology, Rzeszow, Poland (e-mail: k.gomulka@prz.edu.pl).

II. Related Works

The task of weapon detection has seen a variety of research efforts over the years, initially focusing on classical machine learning methods and later shifting towards various deep learning models. Early approaches to weapon detection were focused on handcrafted feature extraction. For example, Żywicki et al. proposed a method using Haar-like features to detect knives while training a cascade classifier [4]. Although the approach incorporated variations in illumination, background, and knife types, its detection performance remained unreliable, which was reflected in a low true positive rate. Another study by Tiwari and Verma [5] proposed a hybrid method that combined color-based segmentation with FREAK descriptors to detect firearms, achieving high initial accuracy. In subsequent work, the authors incorporated SURF features in an effort to improve the robustness to scale and orientation variations, resulting in a noticeable increase in precision. Despite some promising results, classical approaches have become less common due to their high computational cost, limited generalization capabilities, and reliance on domain-specific feature engineering.

To address the limitations of traditional machine learning methods that rely on handcrafted features, weapon detection approaches have started adopting deep learning, particularly convolutional neural networks (CNNs), which have become a cornerstone in the field of object detection [6]. These models enable automatic feature extraction directly from the data, eliminating the need for manual feature engineering and significantly improving detection accuracy. The first CNN-based object detection frameworks were two-stage detectors. In that approach, at first, the model generates region proposals, which are areas that are likely to contain objects. In the second stage, the proposed regions are classified into categories and the bounding boxes are further refined to ensure precise localization. A well-known example of such an architecture is Faster R-CNN [7]. In the context of weapon detection with two-stage detectors, Verma and Dhillon [8] proposed an automatic handgun detection system based on Faster R-CNN and a VGG-16 backbone. The approach used transfer learning and was trained on the Internet Movie Firearms Database, containing firearms in cluttered scenes. Their model demonstrated strong performance in scenarios with occlusion and complex backgrounds. However, reliance on CPU-based training introduced serious computational limitations. Another approach was presented by Pérez-Hernández et al. [9] in which they proposed a hierarchical detection framework using a two-level CNN architecture composed of binary classifiers. The system focused on reducing false positives by isolating the region selection and classification processes.

With the development of more efficient detection pipelines, one-stage detectors such as YOLO [10] and RetinaNet [11] gained popularity due to their faster inference times and simpler architectures. These models combine region proposal and classification into a single step, enabling strong real-time performance. In the case of weapon detection, Salido et al. [1] compared the YOLOv3 and RetinaNet object detectors to identify pistols in surveillance footage. The study investigated whether incorporating information on the posture of individuals holding weapons could reduce false detections. Although the results were promising, the model still produced a significant number of false positives and negatives as a result of the small dataset size and low-resolution input images. Another study focused on a variety of models, including region proposal approaches as well as one-stage detectors such as SSDMobileNetV1, YOLOv3, and YOLOv4. The models were trained on a custom dataset of real-time surveillance CCTV weapon detection [12]. In the end, YOLOv4 was identified as the most effective outperforming other tested options.

Building upon the limitations of convolutional neural network detectors, recent research has explored transformer-based architectures, which have demonstrated strong performance in various object detection tasks due to their ability to model global context and handle complex spatial relationships. Unlike CNN-based models, transformers process the entire image as a sequence, enabling them to capture long-range dependencies more effectively. In the domain of weapon detection, these capabilities have shown promising results. For example, a study by Rodrguez-Ortiz et al. proposed the use of RT-DETR [13] for the detection of handguns in surveillance systems [14]. The model leveraged a hybrid encoder to ensure both high speed and accuracy when deployed on the Nvidia Jetson AGX Xavier, demonstrating its suitability for real-time edge applications. Another work explored the use of the Swin Transformer in combination with models such as Mask R-CNN and Cascade Mask R-CNN to detect a variety of weapons, including pistols, rifles, and knives [15]. Although there were still some limitations to the data, they determined that transformer architectures have the potential to perform better than conventional CNNs and, therefore, to be useful in complex, limited-resource, or challenging contexts.

III. Methodology

Weapon detection in the real-world presents a complicated challenge relating to size, occlusion, background noise, and the visual similarity of objects such as concealed knives or firearms to ordinary everyday items. The presented approach attempts to address the challenges of detection by using multiple datasets with varying weapon examples, as well as using convolutional and transformer models to detect weapons. The aim of this work is to evaluate the usability of state-of-the-art models and explore possible means of their improvement.

A. Dataset

The dataset used consists of combined weapon-related images sourced from multiple datasets. Here, Roboflow datasets were used [16]–[18], as well as the Soha dataset that focuses on weapons and similar looking objects [19]. Since some datasets included more specific weapon categories, such as shotguns, rifles, or knife types, these categories were grouped

into three main classes: gun, knife, and grenade. The merger of weapon types used facilitates analysis by focusing on a limited number of relevant classes. Furthermore, narrowing the scope helps to reduce complexity during model training and evaluation and allows for more targeted research. After that, to ensure that the datasets do not contain duplicate or similar images, a similarity comparison was performed, discarding excessive images.

The dataset comprises 20,969 unique images. The gun class is the largest, reflecting a wider variety of weapon types compared to the other classes. Images originate from diverse sources such as surveillance footage, video and movie frames, and photographic collections, providing varied contexts and visual conditions, as shown in Fig. 1, including:

- surveillance camera footage - simulating real-time monitoring scenarios;
- video stills and movie frames - both from publicly available recordings and cinematic scenes;
- photographic images - such as those taken from public image databases or individual photographs.

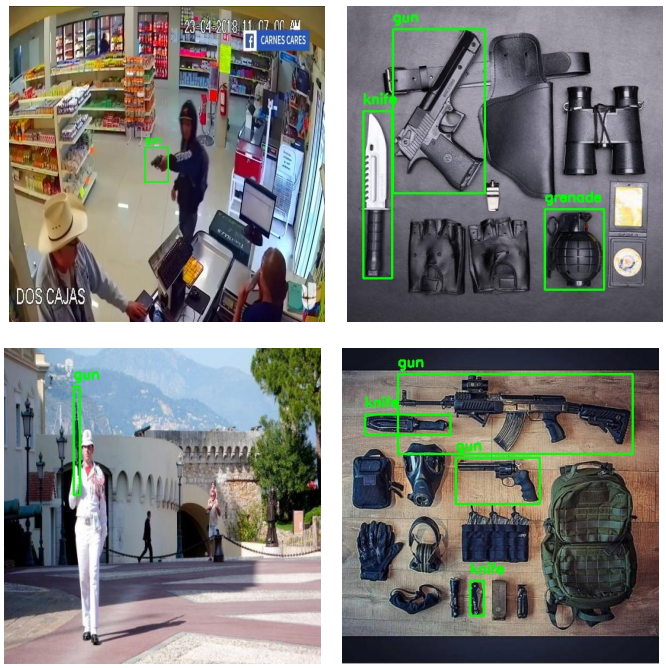


Fig. 1. Examples from the combined dataset showing diverse weapons images

The image diversity ensures that the models are exposed to a wide variety of lighting conditions, viewpoints, and resolutions that are crucial to training a robust object detection model. To prepare the dataset for training, it was divided into training, validation, and testing subsets. Table I provides a detailed breakdown of the distribution of images and weapon instances between training, validation, and test splits, highlighting the number of images, as well as the counts of individual instances for each weapon class. From the data, it is clear that the gun class is the most numerous in the dataset in terms of both the number of instances and the number of images containing guns. This reflects the greater diversity within this class, which includes a wide range of firearms such as pistols, rifles, and shotguns. The knife class, although less numerous than guns, is still well represented, providing substantial examples for model training. The distribution is consistent across the training, validation, and test splits, maintaining a good balance between classes, which is important for reliable model evaluation.

TABLE I
DATASET STATISTICS BY SPLIT AND CLASS

Split	Total Images	Class Images	Class Instances
Train	15,217	Gun: 8,765 Knife: 3,579 Grenade: 1,861	Gun: 11,587 Knife: 4,035 Grenade: 3,210
Validation	3,805	Gun: 2,224 Knife: 896 Grenade: 464	Gun: 3,077 Knife: 1,028 Grenade: 685
Test	1,947	Gun: 1,229 Knife: 443 Grenade: 199	Gun: 1,675 Knife: 482 Grenade: 317
Total	**20,969**	**Gun: 12,218** **Knife: 4,918** **Grenade: 2,524**	**Gun: 16,339** **Knife: 5,545** **Grenade: 4,212**

B. Detection Models

To effectively address the challenges of weapon detection under diverse conditions, a set of object detection models representing different architectures was evaluated. The selection includes convolutional neural networks (CNNs), transformer-based models, and a hybrid approach.

Among CNN-based approaches, models from the YOLO family were selected due to their proven effectiveness in real-time object detection tasks. In particular, YOLOv10 [20] and YOLOv12 [21] were included as state-of-the-art representatives, offering different trade-offs between inference speed and detection accuracy. Each of them can be further selected based on model size. During experiments, nano and small variants for both versions were used. YOLOv10 combines high accuracy with efficiency through a novel dual-label assignment strategy during training (one-to-many and one-to-one), while inference relies solely on one-to-one assignment, eliminating the need for Non-Maximum Suppression (NMS). Its architecture includes an enhanced CSPNet backbone and Path Aggregation Network for multi-scale feature fusion, alongside modules such as partial self-attention and compact inverted bottlenecks to improve feature representation and reduce computation. YOLOv12 advances this design by incorporating an Area Attention module, efficiently approximating global attention on segmented feature maps, and residual ELAN connections to stabilize training and improve gradient flow.

Transformers have also proven to be effective in object detection tasks. The ability to capture global context and therefore improve detection accuracy is crucial, especially in

complex scenarios. The RT-DETR model is an example of transformer-based architecture specifically designed for real-time object detection [13]. Unlike CNN-based models, RT-DETR employs a transformer encoder-decoder structure to process images. First, the backbone extracts feature maps from the input image. After that, the encoder processes the said features, trying to capture contextual relationships, while the decoder uses this information to predict bounding boxes and class labels. RT-DETR proves to be effective in capturing the contextual relationships between different objects within the image. However, the model is more computationally demanding than its CNN-based counterpart, which can be problematic, especially for real-time deployment. For the tests performed, RT-DETRv2 was used [22]. It is an enhanced version that introduces several improvements. RT-DETRv2 optimizes the training strategy by incorporating a deformable attention module with a specified number of sampling points for different feature scales. Allows for more efficient multiscale feature extraction. Additionally, it replaces the original grid sample operator with an optional discrete sampling operator. As a result, typical deployment constraints for DETR-based detectors are removed.

The last chosen model is DETR with YOLO (DEYO), a hybrid approach that combines the strengths of both convolutional and transformer-based models [23]. DEYO builds upon the YOLOv8 architecture by integrating a transformer layer into its detection pipeline. This fusion aims to combine the speed and efficiency of CNNs with the global context awareness provided by transformers. DEYO employs a modified YOLO backbone that is enhanced by a transformer decoder, enabling the model to refine its detection predictions based on global context rather than relying solely on local features.

C. Model Fine-tuning

The models were fine-tuned using the dataset introduced during the experimentation. The fine-tuning process applies pretrained models that were trained on large-scale datasets with general object categories, to the unique visual features of weapons. As a result, the models become more effective in recognizing these specific classes and can also be extended to detect entirely new ones. All training was performed on a system running Ubuntu 22.04, equipped with an AMD Ryzen Threadripper 3970X, 64 GB of RAM, and a single NVIDIA RTX 2080 Ti GPU.

The YOLOv10, YOLOv12 and DEYO models were fine-tuned using the Ultralytics framework with its default training pipeline. Pre-trained models were trained for weapon-detection task based on a custom dataset. Training was conducted for 30 epochs with an input image size of 640×640, a batch size of 8, and a learning rate of 0.001. AdamW was used as an optimizer. In addition, data augmentation was used with operations such as mosaic, random translation, scaling, and color jitter. Some more aggressive augmentations were discarded to preserve the visual characteristics of the weapons.

The RT-DETRv2 model was trained for 30 epochs with an input size of 640×640 and an effective batch size of 8. AdamW optimizer was used with a base learning rate of 0.0001, employing different learning rates for the backbone and normalization layers to better adapt the training. Mixed precision training (AMP) and exponential moving average (EMA) with a decay rate of 0.9999 were used to improve training stability and convergence. To enhance the model, data augmentations were applied during training. The data augmentations included photometric distortions, zoom-out, IoU based cropping, random horizontal flips, and multiscale resizing.

D. Architecture Modifications

Given that RT-DETRv2 achieved the highest performance, this study aims to explore potential enhancements by altering its architecture and loss function. This included examining the effect of the choice of the backbone on the detection performance. Four different backbones were used and tested. ResNet-18 and HGNetv2, which are commonly used in related work and were tested in original RT-DETR paper, as well as two additional architectures, EfficientNetV2 [24] and ConvNeXt [25].

In addition to backbone modification, the loss functions used during training were also adjusted. The original RT-DETR model uses three main types of loss:

- **classification loss** – namely the Variational Focal Loss (VFL), which measures the difference between the probabilities of the predicted class and the true labels;
- **box regression loss** – quantifies how accurate the predicted bounding box coordinates were compared to ground truth, using as default L1 loss;
- **IoU-based loss** – evaluates the overlap between predicted and ground truth bounding boxes, by default using Generalized Intersection over Union (GIoU).

Experiments with various loss functions were conducted to determine which best balances classification and localization accuracy. The main goal was to improve the regression of the bounding box and the overall detection precision. Before exploring IoU-based losses, some standard regression losses were evaluated, including:

- **L1 loss** – penalizes the absolute difference between predicted and true bounding box coordinates:

$$\mathcal{L}_{\text{L1}} = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \,, \tag{1}$$

where $\hat{y}_i$ is the predicted value, y_i is the ground truth, and N is the number of coordinates;

- **L2 loss** – penalizes the squared difference between predicted and true values, placing more emphasis on larger errors:

$$\mathcal{L}_{\text{L2}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \,; \tag{2}$$

- **Huber loss** – a loss that behaves like L2 when the error is small and like L1 when the error is large, making it less sensitive to outliers:

$$\mathcal{L}_\delta(a_i) = \begin{cases} \frac{1}{2}a_i^2 & \text{if } |a_i| \le \delta \\ \delta\left(|a_i| - \frac{1}{2}\delta\right) & \text{otherwise,} \end{cases} \quad \text{with } a_i = \hat{y}_i - y_i, \tag{3}$$

where δ is a threshold parameter, controlling the point where the loss transitions from quadratic to linear.

Experiments with various loss functions were carried out to determine which is best suited for training. The main goal was to improve the regression of the bounding box and the overall detection precision, especially in challenging scenarios. Originally, GIoU was used, which is defined as:

$$\mathcal{L}_{\text{GIoU}} = 1 - \text{IoU} + \frac{|C \setminus (B \cup B_{gt})|}{|C|}, \tag{4}$$

where B and B_{gt} denote the predicted and ground truth bounding boxes, respectively. The term C represents the smallest enclosing box that contains both B and B_{gt}, and IoU is the standard Intersection over Union, calculated as the ratio of the intersection area to the union area of the two boxes.

Instead of GIoU loss, alternatives were tested. First, DIoU (Distance Intersection over Union) [26], which refines the GIoU by taking into consideration the distance between the centers of the predicted and ground truth bounding boxes. It penalizes predictions whose centers are far apart, which can accelerate convergence during training. DIoU is defined as:

$$\text{DIoU} = \text{IoU} - \frac{d^2(\text{c})}{r^2}, \tag{5}$$

where $d(\text{c})$ is the Euclidean distance between the centers of the two boxes, and r is the diagonal length of the smallest enclosing box covering both predicted and ground truth boxes.

Another option, Efficient Intersection over Union (EIoU) [27], extends DIoU by using penalties for differences in the width and height of the bounding boxes. This makes the loss more sensitive to changes in the shape and size of the box. The formula for EIoU is as follows:

$$\text{EIoU} = \text{IoU} - \left(\frac{d^2(\text{c})}{r^2} + \frac{(w_p - w_{gt})^2}{w_{en}^2} + \frac{(h_p - h_{gt})^2}{h_{en}^2}\right), \tag{6}$$

where w_p, h_p are the width and height of the predicted box, w_{gt}, h_{gt} are those of the ground truth, and w_{en}, h_{en} represent the width and height of the enclosing box. By penalizing shape mismatches, EIoU helps improve bounding box regression beyond center alignment.

Spatial Intersection over Union (SIoU) [28] enhances localization loss by combining distance, shape, and angle differences between bounding boxes. By introducing an angular component, it penalizes rotations or skewed predictions relative to the ground truth, improving the localization of irregular objects. The loss can be expressed as:

$$\text{SIoU} = \text{IoU} - \left(\alpha \cdot \frac{d(\text{c})}{r} + \beta \cdot \text{shape} + \gamma \cdot \text{angle}\right), \tag{7}$$

where α, β, γ are weighting factors for center distance, shape difference, and angle difference, respectively. This composite metric improves both the alignment of position and orientation.

Fused Intersection over Union (FIoU) [29] is an IoU-based metric designed to improve the regression of the bounding box by incorporating both the overlap ratio and spatial alignment between the predicted and ground truth boxes. It integrates a normalized distance penalty - ℓ_2, and the squared diagonal length of the smallest enclosing box - ρ^2, leading to a more informative similarity measure. The metric is defined as:

$$\text{FIoU} = \text{IoU} - \frac{\ell_2}{\rho^2}. \tag{8}$$

E. Evaluation Metrics

The model accuracy was evaluated using metrics commonly used in object detection, with the primary one being mean Average Precision (mAP). The chosen metric measures how well a model balances precision and recall at different confidence thresholds. Precision can be defined as the ratio of correctly identified detections in relation to the total number of detections. Recall, on the other hand, refers to the proportion of true positive detections identified by the model, as shown in Equation 9. *TP* (True Positives) represent the number of correctly predicted objects, *FP* (False Positives) the number of incorrect predictions where no object is present, and *FN* (False Negatives) the number of cases where objects were missed.

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}. \tag{9}$$

The area under the precision recall curve is calculated to obtain the Average Precision (AP), which represents the overall precision and recall captured by the model:

$$\text{AP} = \int_0^1 \text{Precision}(\text{Recall})\, d\text{Recall}. \tag{10}$$

The AP has several variants depending on the chosen IoU threshold, which defines how much overlap between the predicted and ground truth boxes is required to consider a detection as correct. The first and most commonly used version is to choose a range of IoU thresholds (between 0.5 and 0.95 with a 0.05 step) and calculate the average. The value of multiple IoU thresholds works to add additional nuance to the AP as it evaluates the model through multiple lenses for IoU overlap in their predicted bounding boxes versus ground truth boxes. In addition to the overall AP across multiple IoU thresholds, two specific cases are often reported, AP50 and AP75, specifically to characterize the models' average precision at fixed IoU thresholds of IoU equal to 0.5 and 0.75, respectively. In doing so, context is maintained for the average precision of the model under moderate restrictions.

The evaluation process can also consider the categories of object size: small, medium and large—represented by the metrics AP_S, AP_M, and AP_L. It is based on weapons detected being classified by the size of their bounding boxes. By computing

the AP independently for the object size categories, additional interpretation of the performance of the model across the object scale can be provided. This interpretation might be important to show the model's strengths and weaknesses in detection, where weapons were presented at differing sizes based on the distance of the camera or angle.

Additionally, in multi-class detection scenarios, the overall mAP is typically obtained by averaging the AP values across all object classes:

$$\text{mAP} = \frac{1}{C}\sum_{c=1}^{C} \text{AP}_c, \tag{11}$$

where C is the total number of classes and AP_c is the average precision computed for class c.

IV. Experimental Results

A. Detection models

In this section, a comparative evaluation of various deep learning models for weapon detection is presented. The models include lightweight convolutional neural networks (YOLOv10n, YOLOv10s, YOLOv12n, YOLOv12s), transformer-based RT-DETRv2, and a hybrid model referred to as DEYOs and DEYOn, respectively. These models were fine-tuned on the custom dataset described in the previous sections.

Table II summarizes the performance of each model using the average precision both for all classes as a mean value and separately for each weapon category. The results indicate that RT-DETRv2 consistently outperforms all other models across all metrics and weapon categories. It achieves the highest overall mAP of 72.30% and mAP50 of 92.20%, with particularly strong performance in the grenade class. Among YOLO-based architectures, YOLOv12s performs best overall, except the grenade class, where YOLOv12n performs better, as seen in the AP50 value. DEYOn and DEYOs, while based on earlier YOLOv8 variants, still deliver competitive results, especially considering their reduced complexity. Overall, the most difficult out of all three classes, for most tested models, proved to be the knife class, especially considering the more restrictive AP metric.

B. Visualization of Detected Regions

To investigate the decision-making process of the best-performing model, Gradient-weighted Class Activation Mapping (Grad-CAM) was implemented in the RT-DETRv2 model [30]. Grad-CAM enables visual interpretation of predictions by highlighting the most influential regions in the input image that contribute to a final detection. In this case, the Grad-CAM visualizations were extracted from the last convolutional layer of the model's ResNet-18 backbone after fine-tuning on the weapon dataset. Fig. 2 presents selected visualizations for weapon detections made by the RT-DETRv2 model, where each detected object achieved a confidence score above 0.1. Subsequent subfigures illustrate the original image containing multiple weapon instances and Grad-CAM heatmaps corresponding to the detections of a gun, knife, and grenade, respectively. The overlaid activation maps highlight the image regions that the model considers most important when making predictions.

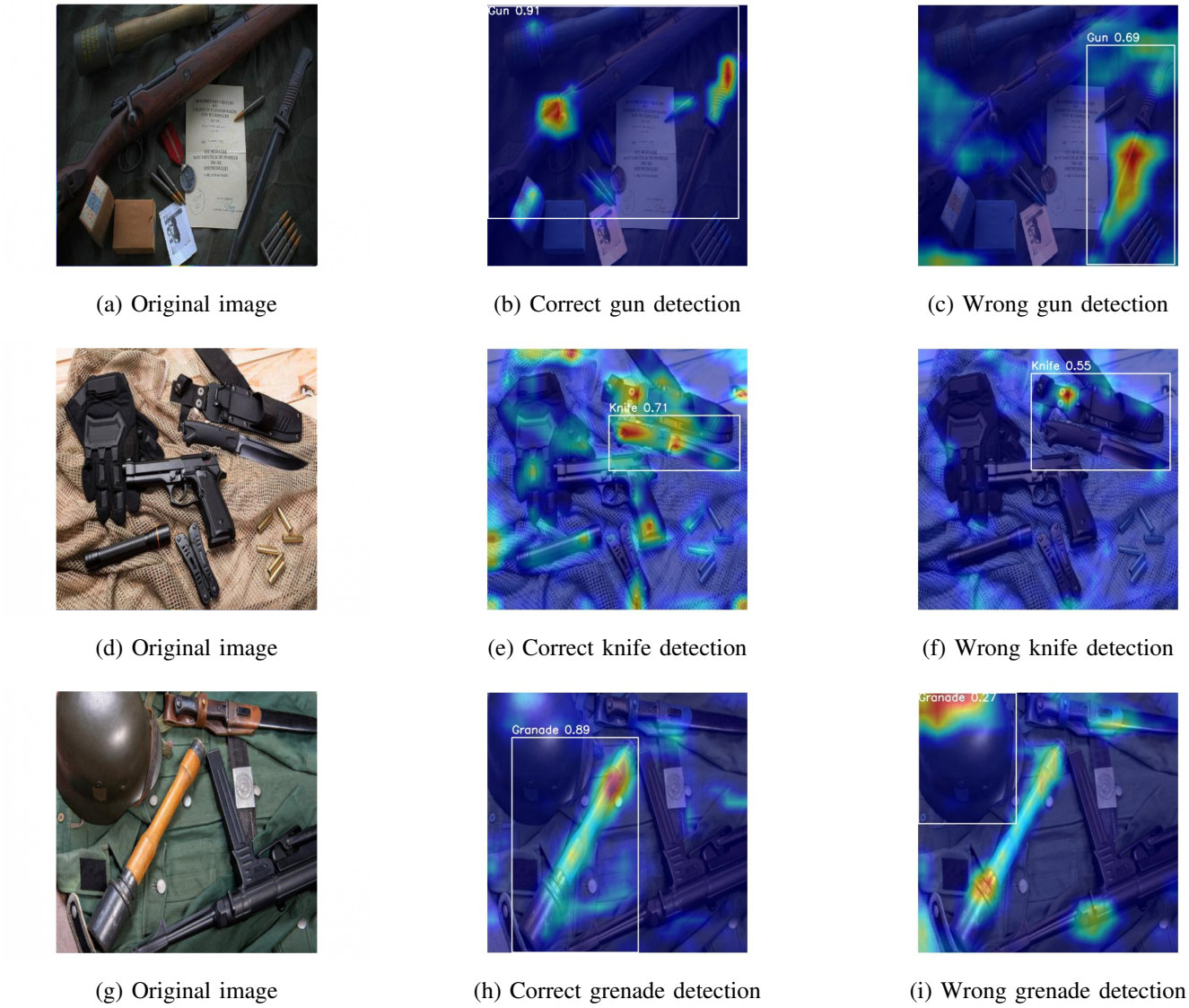
To effectively show how the model makes decisions, three different images were analyzed. In each image, two examples of class are presented, one correct and one incorrect detection. The first three visualizations presented in Figs. 2 (a)–(c) show examples of gun detection. In the case of correct detection, the model focuses on the central portion of the weapon, including the barrel and trigger area, achieving high confidence scores of 0.91. The example shown next to it is less certain and results in detection with lower confidence. In that case, the model incorrectly focuses on the knife blade and makes a wrong prediction by choosing the wrong object. In the second row, the images show that the model successfully identifies the knife, taking into account both the blade and its sheath positioned above it. However, in the incorrect case, only a specific region of the sheath is highlighted, which misleads the model into predicting that area as a knife. Finally, in the last three subfigures, an example of grenade detection is shown. In the correct case, the model clearly highlights the full silhouette of the grenade. But in the incorrect example, although the grenade is still partially visible in heatmap, the model also looks at the surrounding elements, parts of a gun, a knife, and an especially helmet above. The strong influence of the helmet contributes significantly to the final decision, and in the end it is the predicted bounding box returned by the model.

C. RT-DETRv2 Modifications Results

To further enhance detection performance, experiments were conducted with different backbone architectures for the RT-DETRv2 model. In addition, the effect of freezing the backbone weights during fine-tuning was evaluated. Table III presents a comparison of multiple backbone options, including HGNet, ResNet-18, ConvNeXt, and EfficientNet, with and without freezing. The results show that the choice of backbone can affect the final performance. EfficientNet without freezing yields the best overall performance, achieving the highest mAP of 74.60%, outperforming other backbones in large and medium object detection. This suggests that EfficientNet's modern architecture and compound scaling enable more powerful feature extraction when allowed to adapt during training. At the same time, one of the downsides of EfficientNet is the worse result in small object detection presented by the mAPs metric, where ResNet-18 outperforms it. This could be due to ResNet-18's more balanced feature map resolution and inductive biases that benefit small object localization. Experiments also show, that freezing the backbone weights generally leads to decreased performance, especially evident in EfficientNet and ConvNeXt. This emphasizes the importance of the fine-tuning of the backbone of all networks when incorporating architectures not previously adapted to the task domain of RT-DETRv2

TABLE II
COMPARATIVE EVALUATION OF COMPETING DEEP LEARNING MODELS

Model	All		Gun		Knife		Grenade	
	mAP[%]	mAP50[%]	AP[%]	AP50[%]	AP[%]	AP50[%]	AP[%]	AP50[%]
DEYOn	61.02	82.41	62.48	86.35	54.78	83.65	65.80	77.22
DEYOs	62.85	85.46	61.75	85.96	56.74	87.59	70.06	82.83
YOLOv10s	66.72	86.37	67.13	88.30	61.29	87.56	71.73	83.26
YOLOv10n	63.43	83.75	64.51	86.68	56.36	83.19	69.42	81.38
YOLOv12n	66.33	87.33	67.33	88.81	58.24	86.98	73.43	86.21
YOLOv12s	67.30	87.42	68.36	89.59	59.54	86.83	74.01	85.86
RT-DETRv2	**72.30**	**92.20**	**72.20**	**92.82**	**64.49**	**91.43**	**80.20**	**92.34**



(a) Original image (b) Correct gun detection (c) Wrong gun detection

(d) Original image (e) Correct knife detection (f) Wrong knife detection

(g) Original image (h) Correct grenade detection (i) Wrong grenade detection

Fig. 2. Grad-CAM visualizations from RT-DETRv2 for detected objects.

Based on the result previously analyzed, Table IV focuses on the impact of different regression loss functions. All experiments were carried out using the same RT-DETRv2 architecture with a ResNet-18 backbone. The Huber loss achieves the highest overall mAP and mAP75 values. This can be explained by its hybrid behavior. For small errors it acts quadratically, enabling fast and stable convergence, while for larger errors it switches to linear behavior, which reduces the influence of outliers. This combination allows the model to balance precise localization and robustness, preventing large errors from dominating the training process. The L1 loss, which penalizes errors linearly regardless of their size, shows slightly lower

TABLE III
INFLUENCE OF DIFFERENT BACKBONES ON THE PERFORMANCE OF RT-DETRV2

Model	Backbone	Freeze	mAP[%]	mAP50[%]	mAP75[%]	mAP_S[%]	mAP_M[%]	mAP_L[%]
RT-DETRv2	HGNet	✓	70.50	90.70	77.00	29.80	59.40	75.60
RT-DETRv2	HGNet	×	70.20	90.30	76.70	28.40	59.50	75.50
RT-DETRv2	ResNet-18	✓	72.31	91.98	78.76	**33.31**	60.30	77.00
RT-DETRv2	ResNet-18	×	72.30	92.20	78.52	32.70	59.70	76.00
RT-DETRv2	ConvNeXt	✓	65.02	86.03	70.33	26.99	58.31	69.78
RT-DETRv2	ConvNeXt	×	71.40	92.20	77.70	27.00	58.30	77.10
RT-DETRv2	EfficientNet	✓	26.00	44.09	25.96	10.27	14.98	30.13
RT-DETRv2	EfficientNet	×	**74.60**	**93.10**	**80.90**	30.70	**62.20**	**80.20**

overall performance but achieves the best AP50 for the gun class. This suggests that L1 encourages consistent localization that performs well at looser IoU thresholds, focusing on stable detection rather than finely tuned bounding boxes. This effect is noticeable for classes with a larger number of bigger objects like guns. Such behavior can be beneficial when the object appearance is relatively large or well-defined. On contrary, the L2 loss penalizes errors quadratically, strongly emphasizing larger errors. This sensitivity can lead to aggressive correction of mislocalized bounding boxes, which can be advantageous for smaller and more difficult objects, such as grenades. The higher AP75 scores in this category indicate that L2 loss pushes the model to refine bounding boxes more precisely at stricter overlap requirements. However, this can also make the training more sensitive to noisy labels or outliers.

Table V presents a comparison of different IoU-based loss functions used for the bounding box regression. The results show a very similar performance with mAP values around 72.3%. In particular, GIoU achieved the highest AP50 for the gun class (92.82%), while FIoU achieved the best AP (72.27%) and AP75 (79.19%) for the same class. For the knife category, SIoU outperformed other loss functions across all thresholds. In the case of grenade detection, SIoU again achieved the best results in AP50, while EIoU obtained the highest AP (80.31%) and AP75 (84.83%). The DIoU loss function performed the weakest in this evaluation, with a global mAP of only 66.08% and significantly lower values across all object categories. Overall, the results suggest that IoU-based losses such as SIoU and FIoU can significantly improve localization accuracy, particularly for challenging object classes like a knife and a grenade.

Table VI compares different SIoU loss configurations for the bounding box regression with variations in focal loss parameters α and γ, and the weighting of the loss components. The parameters α and γ influence how the model focuses on difficult examples — a higher γ makes the loss concentrate more on difficult samples, while α adjusts the balance between positive and negative examples.

The results indicate that increasing the value of α leads to the highest observed mAP50 score of 72.52%, although only by a small margin. This suggests a limited but measurable influence of the focal loss parameters. Furthermore, assigning too much weight to the IoU loss component results in a slight decrease in both the overall mAP and the mAP50 scores. This highlights the importance of maintaining a balanced contribution between the two components of the bounding box regression loss. Overall, while the observed differences are relatively small, the findings suggest that careful tuning of loss weights and focal parameters can still offer marginal improvements in detection accuracy.

V. CONCLUSIONS

The experiments carried out demonstrate that fine-tuned object detection models can achieve high accuracy in recognizing specific weapon classes. Among the methods evaluated, the transformer-based detector RT-DETRv2 yielded the most promising results, surpassing the performance of its convolutional counterparts. This outcome aligns with recent trends emphasizing the effectiveness of transformer architectures in complex visual recognition tasks, thanks to their global attention mechanisms and enhanced feature representation capabilities.

Grad-CAM-based analysis provided valuable interpretability by highlighting the most influential image regions that contribute to the model predictions. This analysis helped identify potential sources of incorrect detections, such as confusing background clutter or the effect of visually similar objects. By focusing on what the model specifically sees, we could further enhance its ability to distinguish fine-grained visual details by incorporating additional training images that help differentiate commonly mistaken objects, such as helmets misclassified as grenades. This approach could effectively mitigate some of the misclassifications.

Further improvements in detection accuracy were achieved by modifying the best-performing model, specifically by replacing the backbone network with more powerful architectures and experimenting with alternative regression loss functions tailored to bounding box prediction. These adjustments underscore the importance of architectural design choices and loss function optimization in maximizing detection performance.

Future work may involve expanding the dataset to include a broader range of weapon categories and explicitly addressing the detection of concealed items. Moreover, enhancing model

TABLE IV
INFLUENCE OF DIFFERENT LOSS FUNCTIONS ON THE PERFORMANCE OF RT-DETRV2 WITH RESNET-18

Loss Function	All			Gun			Knife			Grenade		
	mAP[%]	mAP50[%]	mAP75[%]	AP[%]	AP50[%]	AP75[%]	AP[%]	AP50[%]	AP75[%]	AP[%]	AP50[%]	AP75[%]
L1 Loss	72.30	92.20	78.52	**72.20**	**92.82**	78.20	64.49	91.43	73.08	80.20	92.34	84.58
Huber Loss	**72.51**	92.18	**79.29**	72.05	92.67	**78.84**	**64.92**	91.64	**73.73**	80.57	92.22	85.31
L2 Loss	72.27	**92.39**	79.24	71.84	92.74	78.74	64.32	**91.67**	72.91	**80.65**	**92.75**	**86.07**

TABLE V
INFLUENCE OF DIFFERENT IOU LOSS FUNCTIONS ON THE PERFORMANCE OF RT-DETRV2 WITH RESNET-18

Loss Function	All			Gun			Knife			Grenade		
	mAP[%]	mAP50[%]	mAP75[%]	AP[%]	AP50[%]	AP75[%]	AP[%]	AP50[%]	AP75[%]	AP[%]	AP50[%]	AP75[%]
GIoU	**72.30**	92.20	78.52	72.20	**92.82**	78.20	64.49	91.43	73.08	80.20	92.34	84.58
DIoU	66.08	89.80	71.72	67.18	91.35	74.20	56.98	88.90	63.19	74.10	89.15	77.77
EIoU	72.01	92.01	**78.75**	71.67	92.57	78.63	64.03	91.20	72.77	80.31	92.26	**84.83**
FIoU	**72.30**	91.17	78.38	**72.27**	92.14	**79.19**	64.55	90.23	72.25	80.08	91.14	83.69
SIoU	72.23	**92.26**	78.67	71.69	92.38	77.69	**64.68**	**91.69**	**73.59**	**80.32**	**92.71**	84.76

TABLE VI
INFLUENCE OF DIFFERENT LOSS CONFIGURATIONS FOR SIOU ON THE PERFORMANCE OF RT-DETRV2 WITH RESNET-18

Loss VFL Params		Loss Weights			All			Gun			Knife			Grenade		
α	γ	w_{vfl}	w_{bbox}	w_{siou}	mAP[%]	mAP50[%]	mAP75[%]	AP[%]	AP50[%]	AP75[%]	AP[%]	AP50[%]	AP75[%]	AP[%]	AP50[%]	AP75[%]
0.5	2.0	1	5	2	72.29	92.08	78.62	71.95	92.47	78.24	64.75	91.38	73.39	80.18	92.39	84.23
0.5	3.0	1	5	2	71.96	90.97	77.88	71.97	92.19	78.56	63.56	89.18	70.54	80.34	91.56	84.54
0.9	2.0	1	5	2	72.42	92.08	79.12	72.15	**92.86**	**79.05**	64.54	90.90	73.43	80.57	92.48	84.87
0.75	3.0	1	5	2	**72.52**	91.52	79.54	**72.30**	92.39	78.89	64.80	90.50	73.01	80.47	91.67	**86.73**
0.75	2.0	1	5	2	72.23	92.26	78.67	71.69	92.38	77.69	64.68	**91.69**	73.57	80.32	**92.71**	84.76
0.75	2.0	1	1	1	72.27	**92.46**	78.72	71.68	92.55	78.31	64.10	91.43	72.22	**81.04**	93.42	85.64
0.75	2.0	1	1	5	72.38	91.43	**79.85**	71.89	92.42	78.97	64.65	90.64	**74.41**	80.59	91.25	86.18
0.75	2.0	1	2	5	72.31	91.46	79.10	71.90	92.14	78.22	**65.14**	91.06	74.13	79.87	91.19	84.94
0.75	2.0	1	5	5	72.16	91.34	78.37	72.11	92.20	78.77	64.67	91.34	72.89	79.70	90.47	83.46

robustness against false positives caused by visually similar everyday objects and integrating complementary information from multiple backbones or other human detection approaches represent promising directions for further research.

REFERENCES

[1] J. Salido, V. Lomas, J. Ruiz-Santaquiteria, and O. Deniz, "Automatic Handgun Detection with Deep Learning in Video Surveillance Images," *Applied Sciences*, vol. 11, no. 13, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/13/6085

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: doi.org/10.1007/978-3-319-10602-1_48

[3] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, no. 7, p. 1956–1981, Mar. 2020. [Online]. Available: http://dx.doi.org/10.1007/s11263-020-01316-z

[4] M. Zywicki, A. Matiolanski, T. Orzechowski, and A. Dziech, "Knife detection as a subset of object detection approach based on Haar cascades," *Proceedings of 11th International Conference On Pattern Recognition and Information Processing*, pp. 139–142, 01 2011.

[5] R. K. Tiwari and G. K. Verma, "A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector," *Procedia Computer Science*, vol. 54, pp. 703–712, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915014076

[6] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, pp. 57–99, 03 2024. [Online]. Available: https://doi.org/10.1007/s10462-024-10721-6

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[8] G. K. Verma and A. Dhillon, "A Handheld Gun Detection using Faster R-CNN Deep Learning," in *Proceedings of the 7th International Conference on Computer and Communication Technology*, ser. ICCCT-2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 84–88. [Online]. Available: https://doi.org/10.1145/3154979.3154988

[9] J. L. S. González, C. Zaccaro, J. A. Álvarez García, L. M. S. Morillo, and F. S. Caparrini, "Real-time gun detection in CCTV: An open problem," *Neural Networks*, vol. 132, pp. 297–308, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608020303361

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [Online]. Available: doi.org/10.1109/CVPR.2016.91

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss

for Dense Object Detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007. [Online]. Available: doi.org/10.1109/ICCV.2017.324

[12] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," *IEEE Access*, vol. 9, pp. 34 366–34 382, 2021. [Online]. Available: doi.org/10.1109/ACCESS.2021.3059170

[13] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *IEEE/CVF Conference Computer Vision Pattern Recognition (CVPR)*, 2024, pp. 16 965–16 974. [Online]. Available: doi.org/10.1109/CVPR52733.2024.01605

[14] L. A. Bustamante and J. C. Gutiérrez, "Real-Time Handgun Detection Using Transformers on Nvidia Jetson AGX Xavier," in *L Latin American Computer Conference (CLEI)*, 2024, pp. 1–6. [Online]. Available: doi.org/10.1109/CLEI64178.2024.10700426

[15] D. T. Son, N. T. K. Tram, and V. T. Anh, "Weapon detection using swin transformer," in *International Conference on Advanced Technologies for Communications (ATC)*, 2023, pp. 328–333. [Online]. Available: doi.org/10.1109/ATC58710.2023.10318911

[16] yolov7test, "Weapon-detection Dataset," https://universe.roboflow.com/yolov7test-u13vc/weapon-detection-m7qso, Feb 2023, visited on 2025-05-12. [Online]. Available: https://universe.roboflow.com/yolov7test-u13vc/weapon-detection-m7qso

[17] Rapidev WD, "Weapon_detection computer vision project," https://universe.roboflow.com/rapidev-wd/weapon_detection-ah5vj, Oct 2022, visited on 2025-05-12. [Online]. Available: https://universe.roboflow.com/rapidev-wd/weapon_detection-ah5vj

[18] Perception07, "Grenade Dataset," https://universe.roboflow.com/perception07-xuigk/grenade-1jsgf, Sep 2024, visited on 2025-05-12. [Online]. Available: https://universe.roboflow.com/perception07-xuigk/grenade-1jsgf

[19] F. Pérez, S. Tabik, A. Castillo Lamas, R. Olmos, H. Fujita, and F. Herrera, "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance," *Knowledge-Based Systems*, vol. 194, p. 105590, 02 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120300678

[20] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-Time End-to-End Object Detection," *arXiv preprint arXiv:2405.14458*, 2024. [Online]. Available: https://arxiv.org/abs/2405.14458

[21] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," *arXiv preprint arXiv:2502.12524*, 2025. [Online]. Available: https://arxiv.org/abs/2502.12524

[22] W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu, "RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer," *arXiv preprint arXiv:2407.17140*, 2024. [Online]. Available: https://arxiv.org/abs/2407.17140

[23] H. Ouyang, "DEYO: DETR with YOLO for End-to-End Object Detection," *arXiv preprint arXiv:2402.16370*, 2024. [Online]. Available: https://arxiv.org/abs/2402.16370

[24] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[25] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 16 133–16 142. [Online]. Available: doi.org/10.1109/CVPR52729.2023.01548

[26] Zhaohui Zheng and Ping Wang and Wei Liu and Jinze Li and Rongguang Ye and Dongwei Ren, "Distance-iou loss: Faster and better learning for bounding box regression," *arXiv preprint arXiv:1911.08287*, 2019. [Online]. Available: https://arxiv.org/abs/1911.08287

[27] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222009018

[28] Z. Gevorgyan, "SIoU Loss: More Powerful Learning for Bounding Box Regression," *arXiv preprint arXiv:2205.12740*, 2022. [Online]. Available: https://arxiv.org/abs/2205.12740

[29] Y. Sun, J. Wang, H. Wang, S. Zhang, Y. You, Z. Yu, and Y. Peng, "Fused-IoU Loss: Efficient Learning for Accurate Bounding Box Regression," *IEEE Access*, vol. 12, pp. 37 363–37 377, 2024. [Online]. Available: doi.org/10.1109/ACCESS.2024.3359433

[30] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626. [Online]. Available: doi.org/10.1109/ICCV.2017.74