Enhancing Two-Stage Object Detection in Aerial Imagery with a Compact Convolutional Transformer RoI Head

Andrej Hulák[1], Bulat Khaertdinov[1], and Mirela Popa*[1]

[1]Department of Advanced Computing Sciences, Maastricht University, Maastricht, The Netherlands

Abstract

Detecting small objects in high-resolution aerial imagery remains a challenging task due to the extreme class imbalance, dense object layouts, and the information loss incurred by aggressive downsampling in standard detection pipelines. In this paper, we introduce CCTdeT (Compact Convolutional Transformer Detector), a two-stage detector that enhances Faster R-CNN by replacing its conventional Region of Interest (RoI) head with a Compact Convolutional Transformer (CCT). The CCT module preserves spatial details using a lightweight convolutional tokenizer, captures global context via transformer encoder layers, and performs classification and bounding-box regression through attention-based sequence pooling. This hybrid design retains the localization precision of two-stage frameworks while significantly improving small-object feature representation. Evaluated on the challenging VisDrone benchmark, CCTdeT achieves a mean Average Precision (mAP) of 0.276 and a mAP@0.50 of 0.510, outperforming the Faster R-CNN baseline. Moreover, it reduces computational cost by 30% (373.4 $\rightarrow$ 264.9 GFLOPs) and increases inference speed (3 $\rightarrow$ 4 FPS). Detailed per-class analyses confirm the efficacy of our approach, demonstrating that modernizing the RoI head with transformer-based modules can yield substantial gains in small-object detection.

1 Introduction

Drone technology, integrated with advances in deep learning-based object detection, has transformed numerous sectors—including surveillance, disaster response, and environmental monitoring—by enabling efficient analysis of complex scenes [CV18, RHGS15, YALZ21]. However, a critical gap exists between standard object detection benchmarks and the demands of aerial imagery [MGP23]. Unlike conventional object detection datasets such as COCO, which primarily feature large-scale, well-separated objects, drone-captured scenes often contain a significantly larger number of small, densely packed objects, distributed unevenly across high-resolution frames [MGP23].

This difference is quantitatively stark. While COCO images contain an average of 7 annotated objects at a width of approximately 640 pixels, the VisDrone [ZWB+18] dataset used in this study features an average of 53 objects per image, with typical resolutions exceeding 1500 pixels in width [MGP23]. These conditions introduce challenges for conventional object detectors, particularly those optimized for COCO-like scenes, and suggest that accuracy gains on standard benchmarks may not directly translate to performance on aerial datasets [MGP23, NBN+25a].

High-resolution aerial imagery poses a unique challenge for object detectors, requiring a careful trade-off between preserving spatial detail and managing computational load. One-stage methods such as the YOLO family [RF16] tackle this by downsampling aggressively to fit large images into memory, but this often leads to the loss of critical features—especially for small or partially occluded targets [NBN+25b]. Pooling operations and fixed receptive fields in convolutional backbones can further degrade signals for small objects, a limitation widely documented in studies of small object detection [MGP23, NBN+25a].

Two-stage frameworks like Faster R-CNN [RHGS15] address this by first generating region proposals and then applying a dedicated classification and regression head to each region, which typically yields better localization accuracy in cluttered or high-density scenes [NBSA24]. This region-level

*Corresponding author: mirela.popa@maastrichtuniversity.nl

reasoning is particularly effective for aerial data, where object sizes vary dramatically. However, the increased computational cost of two-stage architectures often makes them impractical for real-time or resource-constrained deployments [NBN+25a].

Recent work has sought to reimagine the entire detection pipeline using transformer-based architectures (e.g., DETR [WXLS24]) or hybrid convolution–transformer backbones (e.g., PVT [WXL+21], CvT [WXC+21]). While powerful, these models are typically resource-intensive and complex to train, limiting their applicability in time-sensitive aerial applications.

To address all the aformentioned challenges, we explore architectural strategies that preserve local detail while incorporating global context—key for recognizing small objects in large, cluttered scenes. Our investigation is guided by three core research questions:

1. Can we enhance the feature representation of small objects within each proposal without incurring prohibitive computational overhead?
2. Can we integrate modern architectures into a two-stage framework to fuse fine-grained spatial detail with global reasoning?
3. What trade-offs between accuracy, model size, and inference speed emerge from such a design?

We hypothesize that rethinking the Region of Interest (RoI) head—the final but often overlooked component of two-stage detectors—offers a promising path forward. To this end, we introduce CCT-deT, a two-stage detector that replaces Faster R-CNN's conventional RoI head with a Compact Convolutional Transformer (CCT) module. Our hybrid RoI head processes region proposals through a convolutional tokenizer to retain spatial precision, followed by transformer layers that model long-range dependencies. By innovating at the RoI level rather than altering the backbone or proposal mechanism, we preserve the precision of two-stage detection while improving its robustness and speed, especially for small-object challenges.

Extensive experiments on the VisDrone benchmark demonstrate that CCTdeT outperforms Faster R-CNN and a range of one-stage detectors (including YOLOv8 and YOLOv11) across mAP metrics at multiple IoU thresholds. Notably, CCTdeT improves mAP@0.50 by over 2 percentage points compared to Faster R-CNN, while reducing computational cost by nearly 30%, leading to faster inference. A per-class analysis further reveals strong performance across a wide range of object types, from tiny pedestrians and bicycles to larger vehicles—highlighting the model's versatility in dense aerial scenes.

The key contributions of this work are:

1. **RoI-Level Transformer Integration:** We pioneer the use of a Compact Convolutional Transformer as the RoI head in a two-stage detector, leveraging convolutional tokenization and multi-head self-attention to improve feature encoding for small objects.
2. **Efficiency–Accuracy Trade-off:** We show that CCTdeT improves detection accuracy while reducing FLOPs and parameter count, and increasing inference throughput relative to the Faster R-CNN baseline.
3. **Comprehensive Evaluation:** We conduct a thorough empirical study on the VisDrone dataset, including comparisons to state-of-the-art one-stage architectures and a detailed per-class performance breakdown.

The remainder of this paper is structured as follows. Section 2 reviews related work in object detection paradigms, small-object detection strategies, and convolution–transformer hybrids. Section 3 introduces the CCTdeT architecture in detail. Section 4 outlines the experimental setup. Section 5 presents quantitative results, Section 6 discusses broader implications, limitations, and future directions, and Section 7 concludes the paper.

2 Related Works

2.1 Object Detection Paradigms

Modern object detection methods are generally categorized into two main paradigms: two-stage and one-stage detectors. Two-stage detectors, such as Faster R-CNN (Region-based Convolutional Neural

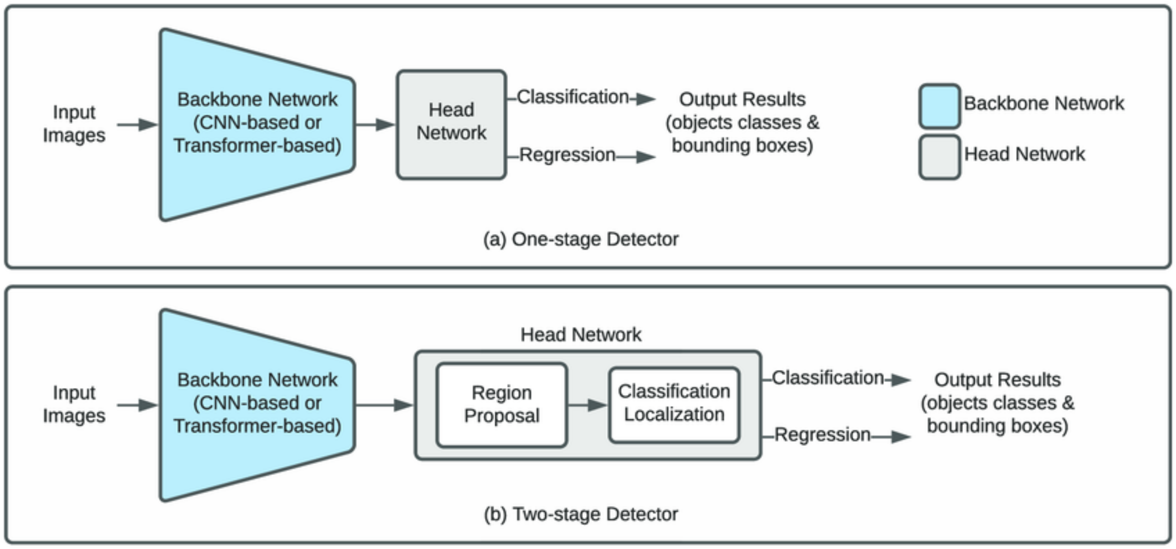

Figure 1: Architectural paradigms of object detection. (a) One-Stage Detector directly performs classification and bounding box regression from features extracted by a backbone network. (b) Two-Stage Detector first generates region proposals via a dedicated component, then refines these proposals through a subsequent classification and localization head [KTOW22].

Network) [RHGS15] and Cascade R-CNN [CV18], first generate candidate regions (Regions of Interest or RoIs) using a Region Proposal Network (RPN) and then classify and refine these proposals in a second stage. This decoupled architecture typically yields higher accuracy, especially in complex or cluttered environments, and remains widely used in domains such as aerial and biomedical imagery [NBSA24].

One-stage detectors, including Single Shot MultiBox Detector (SSD) and the You Only Look Once family (YOLO) [RF16], perform detection in a single forward pass by directly regressing bounding boxes and class scores from predefined anchors or dense spatial locations. These models are computationally efficient and well-suited for real-time applications, though historically they have lagged behind two-stage methods in terms of accuracy—particularly for detecting small or occluded objects [NBSA24, MGP23, RKHD23].

Recent advances have blurred the boundaries between these categories. Anchor-free models like CenterNet and Fully Convolutional One-Stage Object Detector (FCOS) dispense with heuristic anchor boxes by predicting object centers and dimensions directly [NBSA24, NBN+25b]. Furthermore, some anchor-free two-stage detectors—such as Sparse R-CNN [SZJ+20] and Query-Based R-CNN [ZCW+22] variants—have also emerged, treating proposal generation and classification via set prediction frameworks that eliminate dense anchors while maintaining the benefits of region-level processing. Concurrently, architectural hybridization is also contributing to this blurring of boundaries, as hybrid architectures increasingly integrate attention mechanisms with convolutional backbones to combine the benefits of local feature extraction and global context modeling [NBN+25a, NBN+25b, NBSA24].

2.2 Small Object Detection

Small object detection (SOD) presents persistent challenges due to the limited number of pixels such objects occupy [MGP23]. Aggressive downsampling in deep convolutional neural networks (CNNs) often erodes their distinguishing features; additionally, background clutter can easily overwhelm their already weak signals [NBN+25a]. These limitations are particularly severe in high-resolution or densely populated scenes, such as aerial imagery, surveillance footage, or traffic datasets [MGP23].

Several strategies have emerged to address SOD. Feature Pyramid Networks (FPN) [LDG+17] combine spatial detail from shallow layers with semantic depth from deeper ones. Specialized loss functions such as Focal Loss [LGG+17] help rebalance training by focusing on hard examples and underrepresented object sizes. Complementary strategies include super-resolution or upsampling modules that explicitly boost small-object features [NBN+25a, MGP23]. Attention mechanisms—spatial, channel, or both—have also been adopted to highlight likely object regions and suppress distracting backgrounds, yielding substantial gains in small-object average precision [KH24, NBN+25a, WXLS24].

2.3 You Only Look Once Object Detectors

The You Only Look Once family (YOLO) [RF16] has played a central role in the evolution of real-time object detection. The latest versions, such as YOLOv8 [RKHD23] and YOLOv11 [KH24], showcase the trend toward highly optimized, modular architectures. These models use enhanced backbones (e.g., CSPDarknet [WLY+19]), decoupled detection heads, and improved loss functions (e.g., Complete IoU [ZWR+20], Distribution Focal Loss [LGG+17]) to strike a strong balance between speed and accuracy [RKHD23, KH24]. Recent YOLO variants also support multi-scale training and include lightweight versions designed for edge devices or mobile inference. Despite these advances, their reliance on convolutional priors and relatively shallow context modeling can still hamper fine-grained localization and recognition of small, overlapping, or partially occluded objects [NBN+25a].

2.4 Transformers and Hybrid Architectures

Inspired by the success of Vision Transformers (ViTs) [DBK+20] in classification, transformer-based object detectors have emerged as powerful new alternatives. The Detection Transformer (DETR) [WXLS24] introduced a fully attention-based architecture for object detection, treating the task as direct set prediction. While powerful, pure transformer detectors are often compute-intensive and less efficient than CNNs, especially when operating on high-resolution images common in detection, or when trained on smaller datasets [NBN+25a, NBN+25b].

To mitigate some of these issues, hybrid models aim to combine the local inductive biases of convolutional layers with the global context modeling of attention blocks. Architectures like the Convolutional Vision Transformer (CvT) [WXC+21] and Pyramid Vision Transformer (PVT) [WXL+21] integrate transformers into CNN backbones, demonstrating promising results in various vision tasks, including detection. However, even these hybrid backbones, when employed in end-to-end detection pipelines, can still incur substantial computational overhead for real-time or resource-constrained deployment. This is often due to large-scale attention operations across high-resolution feature maps or the inherent complexity of integrating transformers throughout the entire backbone [WSN22, NBSA24, KTOW22]. In parallel, the Compact Convolutional Transformer (CCT) [HWS+21] was proposed as a particularly efficient hybrid, designed to be lightweight while retaining strong representational power, primarily for classification tasks. While CCT has shown promise in classification, its specific application as a dedicated RoI head within a two-stage detector for resource-efficient small object detection remains largely underexplored - and could offer benefits compared to the regular convolutional Faster R-CNN RoI head.

2.5 Motivation and Positioning

Despite the broad advancements in object detection, two-stage detectors like Faster R-CNN continue to dominate in high-precision tasks, particularly when dealing with small, cluttered, or context-sensitive objects [RHGS15,NBN+25a]. While the RPN and backbone designs have received substantial attention over the years [NBSA24], comparatively less work has focused on modernizing the RoI head — the component responsible for classification and bounding-box regression once proposals are generated and pooled.

Our work aims to address this overlooked bottleneck. Unlike transformer-based detectors and many hybrid CNN-transformer backbones that incur significant computational overhead due to large-scale attention operations across entire image features, we propose enhancing the second stage of the two-stage framework. We propose replacing the conventional RoI head with a more compact yet expressive architecture. The goal is to improve representational capacity, especially for small or challenging objects, while keeping inference time and model size competitive. In doing so, we retain the structural advantages of Faster R-CNN — such as region-level reasoning and robustness to object scale variation — but introduce modern design principles that have been underutilized in this part of the pipeline.

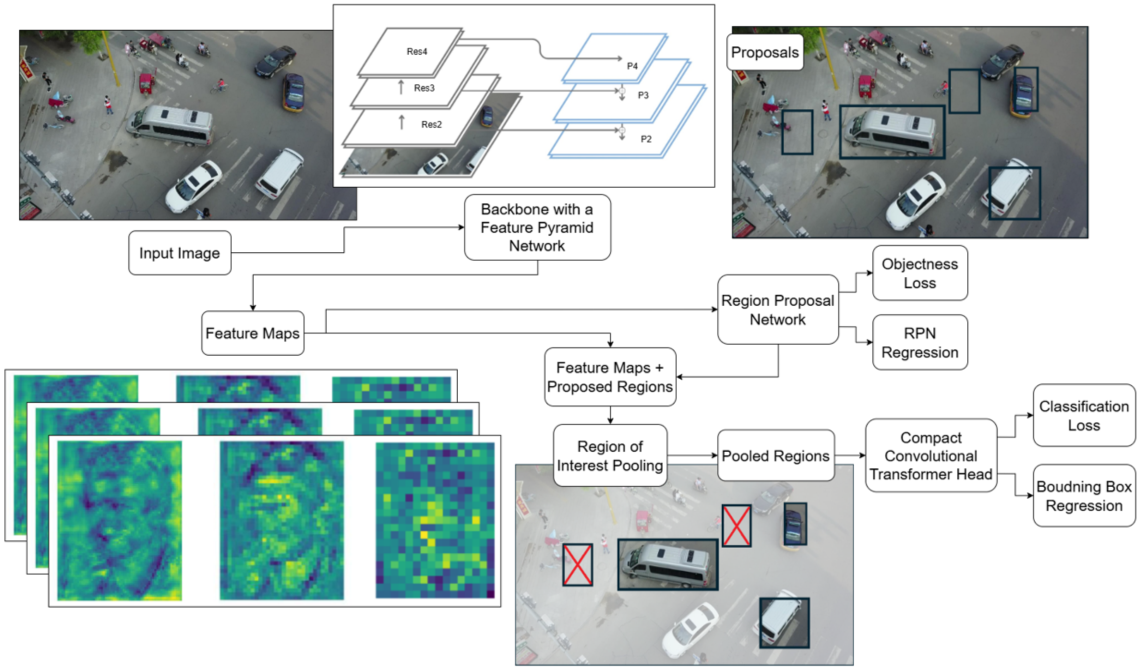

Figure 2: The figure shows the flow of the CCTdeT architecture: input images are processed by a ResNet-50 backbone with FPN to extract multi-scale features. These features go into an RPN to generate candidate object regions. RoI Align pools and resizes proposed regions, which are then passed through a CCT module that outputs predictions for object class and bounding box.

3 Proposed method

We introduce CCTdeT (Compact Convolutional Transformer Detector), a two-stage detector that retains the proven Faster R-CNN pipeline—backbone, feature pyramid, region proposals, and RoI pooling—while replacing its traditional detection head with a Compact Convolutional Transformer (CCT). This section first summarizes the unmodified Faster R-CNN components that serve as our baseline and then details the architecture and integration of the CCT-based RoI head. Finally, we detail our exact implementation choices, including FPN construction, anchor configuration, and CCT hyperparameters drawn from the original CCT ablation studies.

3.1 CCTdeT and its Faster R-CNN Baseline

Our baseline follows the canonical Faster R-CNN pipeline, using a ResNet-50 network pretrained on ImageNet as the feature extractor. We augment this backbone with a five-level feature pyramid to capture information at multiple scales. Before feeding images into the network, we normalize them using the standard ImageNet mean and standard deviation and resize them so that the shorter side measures 800 pixels (while ensuring the longer side does not exceed 1333 pixels).

The Region Proposal Network then operates on each level of the feature pyramid, placing anchor boxes of five distinct scales—approximately 8, 16, 32, 64, and 128 pixels on the shorter side—and three aspect ratios (0.5, 1.0, and 1.5). These scales are determined by k-means clustering on the VisDrone training set's bounding-box dimensions, ensuring that proposals align well with the dataset's object size distribution. For each anchor, the network predicts both an objectness confidence and a set of bounding-box offset adjustments. We keep the top 2,000 proposals by score before applying non-maximum suppression, and the top 1,000 after suppression, in both training and inference.

Each surviving proposal is then cropped from the appropriate feature map and passed through RoIAlign to produce a fixed 7×7 feature patch. In the original Faster R-CNN design, these patches would be flattened and processed by two fully connected layers for final classification and bounding-box regression. In our baseline, we retain this exact configuration to provide a point of comparison for our

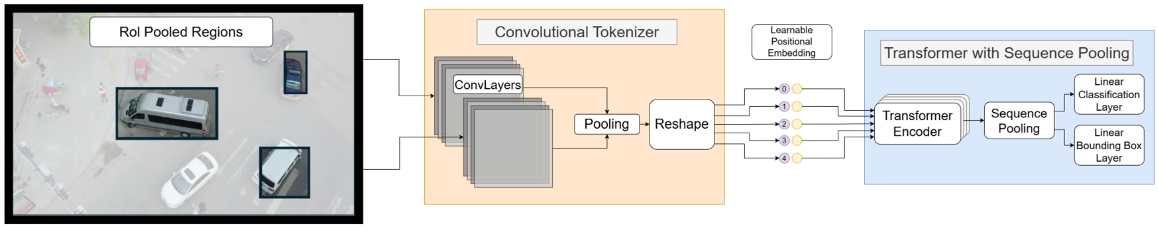

Figure 3: Overview of the CCT-based RoI head architecture integrated into the CCTdeT model. RoI pooled regions are first passed through a convolutional tokenizer that extracts spatial features using convolutional and pooling layers. The resulting feature maps are reshaped into token sequences, augmented with learnable positional embeddings, and processed by transformer encoders. The encoded tokens are then aggregated via sequence pooling, and the resulting compact representation is fed into two parallel linear layers for classification and bounding box regression.

CCT-enhanced RoI head.

3.2 CCT-Enhanced RoI Head

To integrate the CCT into the Faster R-CNN architecture, we replace the conventional RoI head with a compact, attention-based module that processes each pooled RoI feature map as a miniature sequence classification task.

The CCT-based RoI head begins with a convolutional tokenizer comprising two convolutional layers, each with a kernel size of 7, stride of 2, and padding of 3. These layers operate on the RoI-aligned feature maps, preserving local spatial structure while reducing feature dimensionality. The output is then reshaped into a sequence of 49 tokens, where each token is a 384-dimensional vector. Learnable positional embeddings are added to each token to preserve spatial ordering. The resulting token sequence is processed by four transformer encoder layers, each with four attention heads and a feedforward network with an MLP ratio parameter of 3. Instead of using a fixed class token, the model aggregates the sequence using attention-based sequence pooling, which computes a weighted sum over all tokens based on learnable attention scores. This operation produces a single 384-dimensional embedding per RoI. Finally, two lightweight linear layers map this embedding to classification logits and bounding box regression offsets, respectively.

The configuration of the CCT module used in our detection head is based on the ablation studies presented in the original CCT paper [HWS+21], which report that a shallow transformer with four encoder layers and a 384-dimensional learnable embedding offers a strong balance between efficiency and accuracy. We adopt this configuration without modification, demonstrating that it can be effectively transplanted into a two-stage detector to enhance region-level reasoning. Notably, all other components of the Faster R-CNN framework, including anchor generation, RPN scoring, and FPN construction, are preserved in their original form to isolate the impact of the new RoI head.

This integration enables a direct comparison to the baseline Faster R-CNN detector while clearly isolating the benefits introduced by replacing the RoI head with a compact transformer module.

4 Experimental Setup

All models were trained using the Ultralytics [JQC23] deep-learning pipeline to ensure a uniform and reproducible experimental framework. In addition to our proposed CCTdeT model, we trained a suite of comparative detectors under identical conditions: five variants of YOLOv8 (Nano, Small, Medium, Large, and X-Large), five analogous variants of the emerging YOLOv11, and a ResNet-50-FPN Faster R-CNN baseline. Each network was initialized from its official pretrained weights. All models were trained on a single NVIDIA GeForce RTX 4060 GPU with mixed-precision enabled. We set a hard cap of 100 epochs and relied on early stopping based on validation mAP to terminate training once improvements plateaued. The same augmentation, optimization, and scheduling protocols were applied to every model, ensuring a fair and direct comparison across architectures.

Model	Parameters	GFLOPs	FPS	mAP	mAP@0.50	mAP@0.75
YOLOv11n	2.580 M	6.300	40	0.202	0.345	0.200
YOLOv11s	9.417 M	21.300	32	0.246	0.408	0.248
YOLOv11m	20.038 M	67.700	19	0.288	0.464	0.297
YOLOv11l	25.287 M	86.600	17	0.296	0.472	0.308
YOLOv11x	56.839 M	194.500	7	**0.305**	0.486	**0.317**
YOLOv8n	3.008 M	8.100	43	0.195	0.336	0.191
YOLOv8s	11.129 M	28.500	28	0.239	0.401	0.241
YOLOv8m	25.846 M	78.700	22	0.271	0.444	0.276
YOLOv8l	43.614 M	164.900	15	0.280	0.454	0.289
YOLOv8x	68.133 M	257.400	11	0.287	0.462	0.296
CCTdeT	36.046 M	264.900	4	0.276	**0.510**	0.261
Faster R-CNN	43.302 M	373.400	3	0.262	0.491	0.247

Table 1: Performance comparison on VisDrone validation set

We validated each model after every epoch. Final comparisons include mean average precision (mAP) across IoU thresholds, inference speed, parameter count, and computational cost. We also investigate mAP@.50 for all classes individually and provide normalized confusion matrices for CCTdeT, YOLOv11x and Faster R-CNN. It is important to note that all evaluation results were obtained with a single NVIDIA GeForce 1650 GPU.

We optimized all models with the AdamW optimizer, starting from a learning rate of 0.01 and momentum of 0.937. A three-epoch warmup schedule gradually raised both the learning rate and momentum to their target values, and a global weight decay of 0.0005 was applied to all learnable parameters. For the two-stage detectors, objectness and classification branches employed Binary Cross-Entropy (BCE) loss, while both RPN and box-regression heads used smooth L1 loss. The one-stage detectors all used BCE for classification and Complete Interesection over Union loss for bounding box regression.

To improve robustness to the unique challenges of aerial imagery, we applied an extensive data augmentation pipeline. Geometric augmentations—random rotations, translations, scaling, shearing, perspective warps—and both horizontal and vertical flips diversified object poses. Color perturbations in the HSV space simulated variable lighting and sensor conditions. We further incorporated advanced mixing strategies: Mosaic augmentation to merge four images into one, MixUp to interpolate image pairs and their labels, RandAugment for automated policy search, and random erasing to occlude random patches. Finally, Copy-Paste augmentation specifically addressed high object density by moving annotated objects from one image into another.

5 Experimental Results

Having established our experimental setup and methodology, we now present the performance evaluation of CCTdeT against several state-of-the-art detectors on the VisDrone validation set, as summarized in Table 1. Our analysis focuses on mAP (mean Average Precision) at various IoU thresholds, computational efficiency (GFLOPs and Parameters), and inference speed (FPS). Furthermore we also analyze class specific performance of each model at mAP@0.50 and also include the corresponding confusion normalized matrices for CCTdeT, YOLOv11x and Faster R-CNN.

Table 5 show that CCTdeT consistently outperforms the standard Faster R-CNN baseline and offers a compelling alternative to one-stage detectors on the VisDrone validation set. In terms of overall mean Average Precision (mAP) across IoU thresholds, CCTdeT achieves 0.276 compared to Faster R-CNN's 0.262, while also reducing computational cost from 373.4 GFLOPs to 264.9 GFLOPs and parameter count from 43.30 M to 36.05 M. Importantly, this gain in efficiency is accompanied by a one-frame-per-second increase in inference speed (4 FPS versus 3 FPS), demonstrating that the compact transformer

Model	pedestrian	people	bicycle	car	van	truck	tricycle	awning-tricycle	bus	motor	mAP@0.50
YOLOv11n	0.363	0.287	0.096	0.763	0.389	0.317	0.224	0.127	0.502	0.386	0.345
YOLOv11s	0.446	0.341	0.147	0.799	0.443	0.396	0.295	0.154	0.603	0.458	0.408
YOLOv11m	0.510	0.385	0.217	0.831	0.500	0.455	0.365	0.206	0.650	0.523	0.464
YOLOv11l	0.531	0.398	0.233	0.834	0.503	0.472	0.364	0.186	0.663	0.534	0.472
YOLOv11x	0.542	0.412	0.255	0.840	0.522	**0.474**	0.388	**0.206**	0.667	0.551	0.486
YOLOv8n	0.353	0.285	0.088	0.757	0.382	0.302	0.228	0.117	0.484	0.361	0.336
YOLOv8s	0.434	0.339	0.139	0.797	0.455	0.371	0.270	0.156	0.597	0.447	0.401
YOLOv8m	0.488	0.387	0.194	0.815	0.462	0.427	0.337	0.178	0.659	0.498	0.444
YOLOv8l	0.506	0.393	0.196	0.825	0.481	0.438	0.349	0.183	0.657	0.512	0.454
YOLOv8x	0.523	0.397	0.218	0.825	0.490	0.441	0.359	0.188	0.658	0.520	0.462
CCTdeT	**0.618**	**0.469**	**0.297**	**0.857**	**0.544**	0.451	**0.398**	0.204	**0.670**	**0.586**	**0.510**
Faster R-CNN	0.586	0.443	0.266	0.853	0.533	0.446	0.370	0.198	0.652	0.562	0.491

Table 2: Per-class mAP@0.50 on the VisDrone validation set

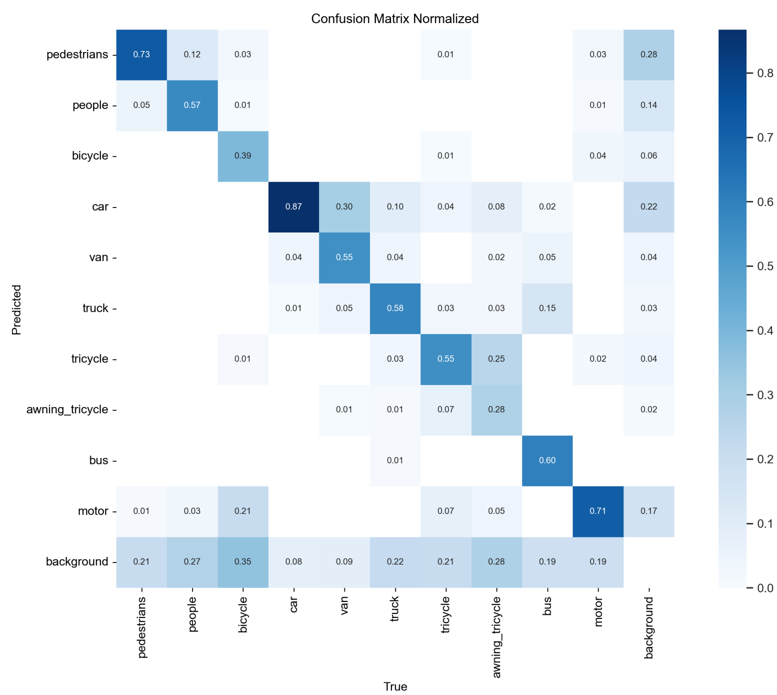

Figure 4: Normalized confusion matrix for CCTdeT at IoU@0.45.

head enhances both accuracy and throughput within the same two-stage framework.

When examining mAP at a more lenient IoU threshold of 0.50, CCTdeT attains the score of 0.510, surpassing all one-stage YOLO variants and the Faster R-CNN baseline. While the largest YOLO model, YOLOv11x, leads in precise localization at mAP@0.75 with 0.317 and overall mAP of 0.305, it does so at the expense of a heavier compute footprint (194.5 GFLOPs) and slower real-time performance (7 FPS). Conversely, the smallest YOLO models, such as YOLOv8n, achieve high frame rates (43 FPS) but only a 0.336 mAP@0.50, reflecting the familiar trade-off between speed and accuracy. CCTdeT bridges this gap by delivering the highest mAP@0.50 among all tested models, making it especially well suited for applications where reliably detecting the presence of objects—rather than perfect boundary alignment—is critical.

A deeper look at class-specific performance (table 5) at mAP@0.50 further highlights CCTdeT's strengths. Across all ten VisDrone categories—ranging from pedestrian and people to complex vehicle classes like awning-tricycle—CCTdeT leads or matches the best per-class scores. It achieves its largest gains over Faster R-CNN on pedestrian (0.618 versus 0.586), people (0.469 versus 0.443), and bicycle (0.297 versus 0.266). Even in challenging classes with small or densely packed instances, such as tricycle and awning-tricycle, CCTdeT attains 0.398 and 0.204 respectively, compared to Faster R-CNN's 0.370 and 0.198. These class-level improvements demonstrate that the CCT-based RoI head

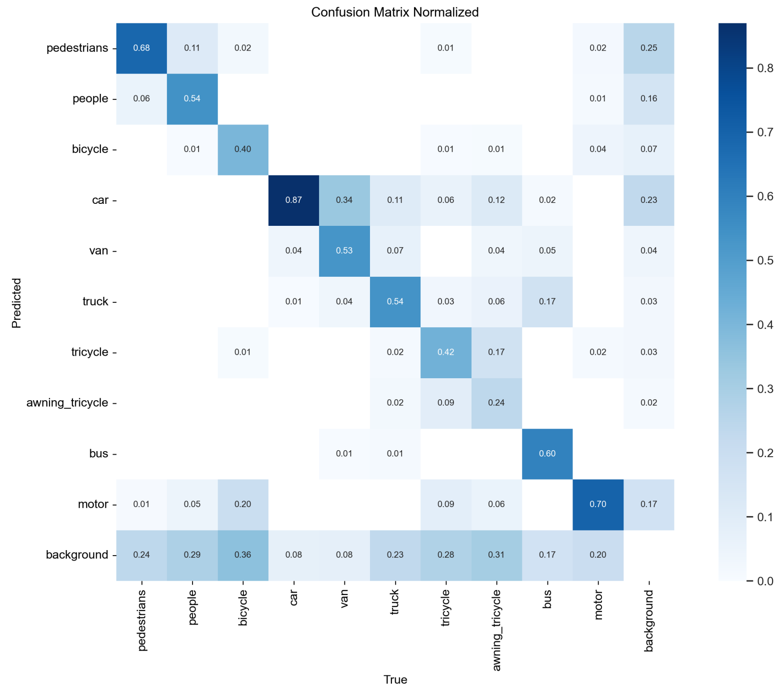

Figure 5: Normalized confusion matrix for Faster R-CNN at IoU@0.45.

enhances region-level reasoning across diverse object types, validating our approach as an effective refinement of the two-stage detection paradigm.

Figures 4, 5, and 6 present the normalized confusion matrices for the proposed CCTdeT model and the baseline Faster R-CNN and YOLOv11x detectors. The CCTdeT model demonstrates superior classification accuracy across several critical object categories, particularly those that are visually similar or small in scale.

Specifically, CCTdeT outperforms both Faster R-CNN and YOLOv11x in distinguishing pedestrians (0.73 vs. 0.68 vs. 0.47) and people (0.57 vs. 0.54 vs. 0.31), indicating more refined separation between overlapping human-centric classes. For bicycle classification, CCTdeT (0.39) achieves a major improvement over YOLOv11x (0.25), while performing comparably to Faster R-CNN (0.40).

In the vehicle categories, all three models demonstrate strong detection of cars, with CCTdeT and Faster R-CNN achieving identical accuracy (0.87), and YOLOv11x slightly lower (0.79). For van and truck categories, CCTdeT again leads with accuracies of 0.55 and 0.58, compared to Faster R-CNN (0.53, 0.54) and YOLOv11x (0.46, 0.42), suggesting enhanced representation of mid-size vehicle features.

Performance on smaller and more ambiguous categories such as tricycles (0.55 vs. 0.42 vs. 0.37) and awning tricycles (0.28 vs. 0.24 vs. 0.18) further highlights the CCTdeT model's robustness, with noticeable reductions in confusion with other vehicle classes. Moreover, motor classification accuracy is slightly improved in CCTdeT (0.71), compared to Faster R-CNN (0.70) and YOLOv11x (0.51), again reflecting better feature discrimination.

Finally, CCTdeT achieves stronger background suppression, with a lower tendency to misclassify foreground objects as background (e.g., 0.22–0.28 background confusion for key classes), compared to higher confusion levels in YOLOv11x (e.g., 0.42–0.59 across multiple classes). This indicates CCTdeT's potential for reducing false negatives in cluttered aerial scenes.

6 Discussion

Our experimental results demonstrate that integrating the CCT model into the RoI head of Faster R-CNN yields meaningful improvements in both accuracy and efficiency for small-object detection in aerial imagery. By replacing the traditional RoI head with a convolutional tokenizer, transformer

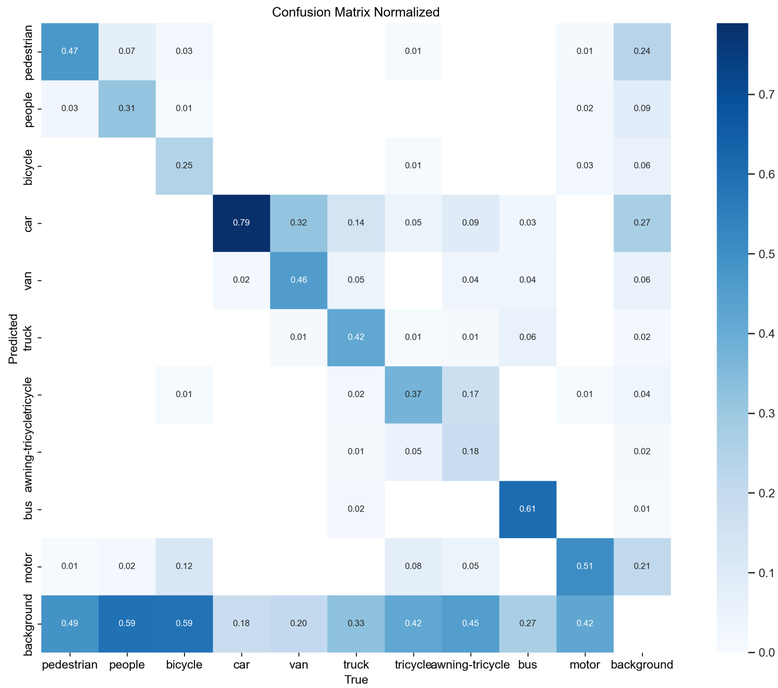

Figure 6: Normalized confusion matrix for YOLOv11x at IoU@0.45.

encoder layers, and attention-based sequence pooling, CCTdeT achieves a higher overall mAP (0.276) than the Faster R-CNN baseline (0.262) while simultaneously reducing computational demand by nearly 30% (from 373.4 to 264.9 GFLOPs) and speeding up inference by one frame per second. These gains underscore that the RoI, often overlooked, presents an opportunity for impactful architectural improvement.

A particularly striking outcome is CCTdeT's leading mAP@0.50 of 0.510, which surpasses all one-stage YOLO variants and the Faster R-CNN baseline. This achievement suggests that the combination of local convolutional processing and global self-attention is especially effective at preserving and integrating the minute spatial cues that characterize small objects. In practical terms, CCTdeT's advantage at a lenient IoU threshold means it excels at reliably flagging the presence of objects even when precise boundary alignment is challenging. This property is highly desirable for applications such as surveillance, search-and-rescue, or environmental monitoring, where detecting the existence of small targets often takes precedence over pixel-perfect localization.

The class-level breakdown further reinforces CCTdeT's versatility. Across ten distinct categories, including pedestrians, people, bicycles, and various vehicle types, CCTdeT either matches or exceeds the best per-class performance achieved by other detectors. Its largest gains appear for pedestrian, people, and bicycle classes, where fine texture and shape features are easily lost by standard pooling and flattening. By contrast, even the strongest one-stage models show clear shortcomings on these smaller, less textured categories. CCTdeT's consistent superiority across all classes indicates that the hybrid convolution-transformer RoI head generalizes well, rather than benefiting only a narrow subset of scenarios.

Several limitations merit further discussion: CCTdeT's inference speed of 4 FPS, while faster than the baseline, remains below the 11–43 FPS achieved by YOLO variants. For latency-critical deployments, further engineering—such as model pruning, quantization, or custom hardware acceleration—would be required to close this gap. Likewise, incorporating advanced regression losses such as CIoU could further tighten boundary localization. We also acknowledge the importance of hyperparameter tuning for model performance. A more exhaustive exploration of the hyperparameter space, possibly using automated tuning techniques, could further enhance the model's accuracy and efficiency even further.

6.1 Future Work

Building upon these findings, our work identifies several promising directions for continued development. First, combining CCTdeT with modern backbone and RPN advances—such as hybrid CNN-transformer backbones or guided anchoring—could magnify its benefits. Second, extending the CCT RoI head to instance segmentation or keypoint detection tasks may prove equally fruitful, given its demonstrated capacity for fine-grained feature encoding.

Furthermore, one can imagine evolving it into a fully end-to-end object detector in its own right. In such an architecture, the convolutional tokenizer would serve as the primary feature extractor, producing a rich sequence of spatial tokens directly from the input image at multiple scales. These tokens could then flow through a hierarchical series of transformer encoder stages, each refining the representation by attending both within and across scales, effectively replacing the traditional backbone and feature pyramid with a unified, attention-driven hierarchy. Rather than relying on an external region proposal network, the model could generate object queries that attend to these token embeddings to predict class labels and box coordinates in a single pass, similar to set-prediction approaches. To keep computation manageable, the transformer depth could be made dynamic, allocating more attention layers only to regions with high uncertainty or complexity while processing simpler areas with lighter encoding. Finally, the same sequence-pooled embeddings that drive classification and localization could feed parallel heads for mask segmentation or keypoint prediction, yielding a unified multitask network.

7 Ablation Study

We extended our previous experiments by performing ablation studies on the CCTdeT model. In this study we use several versions of the CCT head, and observe the effects of changing hyperparameters on metrics such as speed, precision and also overall performance of the model.

8 Conclusion

In this work, we introduced CCTdeT (Compact Convolutional Transformer Detector), a novel two-stage detector that addresses these challenges by modernizing the RoI head of Faster R-CNN with a Compact Convolutional Transformer (CCT). By integrating convolutional tokenization for local detail preservation and transformer layers for global context modeling, CCTdeT enhances feature representation for small objects while maintaining computational efficiency. Our experiments on the VisDrone dataset demonstrate that CCTdeT outperforms the Faster R-CNN baseline by 2.9% in mAP@0.50 (0.510 vs. 0.491) while reducing computational costs by 30% (264.9 vs. 373.4 GFLOPs) and accelerating inference by 33% (4 FPS vs. 3 FPS). The model's hybrid design proves particularly effective for small-object categories, achieving gains of 5.4% for pedestrians and 11.6% for bicycles over Faster R-CNN. These results support our hypothesis that refining the RoI head, rather than redesigning the entire architecture, can significantly improve the balance between accuracy and efficiency.

CCTdeT's strong performance at lenient IoU thresholds makes it well suited for real-world applications like surveillance and environmental monitoring, where detecting the presence of small objects is often more critical than pixel-perfect localization. However, its inference speed (4 FPS) remains a limitation for latency-sensitive deployments compared to lightweight YOLO variants (e.g., YOLOv11x at 7 FPS). Future work could bridge this gap through model compression, dynamic token pruning, or hardware-aware optimizations. Moreover, evaluating CCTdeT on additional datasets, including simpler or cross-domain benchmarks, would further clarify its generalization capability and robustness across diverse visual conditions. By demonstrating that modernizing the RoI head can revitalize two-stage detectors, this work challenges the prevailing focus on backbone and RPN innovation, reaffirming the continued relevance of two-stage architectures amidst the rise of one-stage paradigms. It offers a scalable pathway for aerial object detection in the era of transformers.

Code Availability

The source code for CCTdeT, including the model implementation, training scripts, and evaluation procedures, is publicly available to ensure reproducibility and facilitate further research. The code can be accessed at: https://github.com/andrejhulak/cctdet.

Ablation Studies

CCT Config (ROI Head)	Params (M)	mAP	mAP@0.5	mAP@0.75
Dim=128, Depth=2, Heads=2, MLP=1.0, Conv=2, k=3, s=1	28.5	0.228	0.452	0.203
Dim=128, Depth=4, Heads=2, MLP=1.0, Conv=2, k=3, s=1	28.7	0.255	0.489	0.236
Dim=256, Depth=6, Heads=4, MLP=2.0, Conv=2, k=3, s=1	31.5	0.267	0.501	0.247
Dim=256, Depth=7, Heads=4, MLP=2.0, Conv=1, k=3, s=1	32.4	0.281	0.513	0.271
Dim=256, Depth=7, Heads=4, MLP=2.0, Conv=2, k=7, s=2	33.4	0.264	0.494	0.251
Dim=64, Depth=2, Heads=2, MLP=2.0, Conv=1, k=3, s=1	28.3	0.249	0.478	0.227
Dim=256, Depth=4, Heads=4, MLP=2.0, Conv=1, k=3, s=1	30.8	0.277	0.507	0.262
Dim=384, Depth=2, Heads=4, MLP=2.0, Conv=2, k=7, s=1	32.5	0.268	0.494	0.255
Dim=768, Depth=4, Heads=4, MLP=2.0, Conv=2, k=7, s=2	50.5	0.276	0.501	0.266
Dim=128, Depth=2, Heads=2, MLP=2.0, Conv=2, k=7, s=2	29.5	0.259	0.485	0.245
Dim=256, Depth=2, Heads=2, MLP=1.0, Conv=2, k=7, s=2	30.5	0.258	0.485	0.243
Dim=128, Depth=4, Heads=4, MLP=1.0, Conv=4, k=7, s=2	30.1	0.262	0.494	0.243
Dim=512, Depth=2, Heads=2, MLP=1.0, Conv=2, k=7, s=2	33.8	0.268	0.494	0.253
Dim=512, Depth=2, Heads=2, MLP=3.0, Conv=1, k=7, s=2	39.9	**0.288**	**0.514**	**0.285**
Dim=512, Depth=1, Heads=2, MLP=4.0, Conv=2, k=7, s=2	33.7	0.269	0.500	0.253
Dim=512, Depth=1, Heads=2, MLP=4.0, Conv=2, k=7, s=2	33.7	0.270	0.498	0.256
Dim=512, Depth=1, Heads=1, MLP=3.0, Conv=1, k=7, s=2	37.2	0.280	0.507	0.273
Dim=224, Depth=2, Heads=2, MLP=2.0, Conv=2, k=7, s=2	30.4	0.273	0.503	0.264
Dim=384, Depth=2, Heads=8, MLP=2.0, Conv=2, k=7, s=2	32.5	0.265	0.493	0.247
Dim=512, Depth=2, Heads=1, MLP=2.0, Conv=1, k=7, s=2	38.8	0.284	0.510	0.277
Dim=384, Depth=4, Heads=4, MLP=3.0, Conv=2, k=7, s=2	36.0	0.276	0.509	0.260

Table 3: Compact Convolutional Transformer ROI-head variants on VisDrone validation set.

We have also analyzed the contribution of each architectural component to the final performance, conducting a set of ablation studies. We systematically varied the hyperparameters across the model's main modules to evaluate their impact on both mAP and inference speed. The key areas of investigation are detailed below.

CCT Config (ROI Head)	mAP	mAP$_S$	mAP$_M$	mAP$_L$	FPS
Dim=128, Depth=2, Heads=2, MLP=1.0, Conv=2, k=3, s=1	0.228	0.165	0.304	0.278	3.37
Dim=128, Depth=4, Heads=2, MLP=1.0, Conv=2, k=3, s=1	0.255	0.189	0.334	0.314	3.02
Dim=256, Depth=6, Heads=4, MLP=2.0, Conv=2, k=3, s=1	0.267	0.200	0.350	0.326	1.91
Dim=256, Depth=7, Heads=4, MLP=2.0, Conv=1, k=3, s=1	0.281	0.214	0.362	0.344	1.77
Dim=256, Depth=7, Heads=4, MLP=2.0, Conv=2, k=7, s=2	0.264	0.198	0.348	0.315	3.65
Dim=64, Depth=2, Heads=2, MLP=2.0, Conv=1, k=3, s=1	0.249	0.185	0.328	0.308	3.73
Dim=256, Depth=4, Heads=4, MLP=2.0, Conv=1, k=3, s=1	0.277	0.211	0.356	0.349	2.34
Dim=384, Depth=2, Heads=4, MLP=2.0, Conv=2, k=7, s=1	0.268	0.202	0.346	0.362	2.19
Dim=768, Depth=4, Heads=4, MLP=2.0, Conv=2, k=7, s=2	0.276	0.208	0.357	0.337	3.65
Dim=128, Depth=2, Heads=2, MLP=2.0, Conv=2, k=7, s=2	0.259	0.192	0.340	0.337	3.44
Dim=256, Depth=2, Heads=2, MLP=1.0, Conv=2, k=7, s=2	0.258	0.198	0.337	0.328	3.65
Dim=128, Depth=4, Heads=4, MLP=1.0, Conv=4, k=7, s=2	0.262	0.194	0.345	0.364	3.25
Dim=512, Depth=2, Heads=2, MLP=1.0, Conv=2, k=7, s=2	0.268	0.202	0.346	0.359	3.65
Dim=512, Depth=2, Heads=2, MLP=3.0, Conv=1, k=7, s=2	**0.288**	**0.222**	**0.370**	**0.356**	3.13
Dim=512, Depth=1, Heads=2, MLP=4.0, Conv=2, k=7, s=2	0.269	0.204	0.346	0.354	3.81
Dim=512, Depth=1, Heads=2, MLP=4.0, Conv=2, k=7, s=2	0.270	0.205	0.351	0.361	3.90
Dim=512, Depth=1, Heads=1, MLP=3.0, Conv=1, k=7, s=2	0.280	0.216	0.363	0.357	3.25
Dim=224, Depth=2, Heads=2, MLP=2.0, Conv=2, k=7, s=2	0.273	0.207	0.352	0.362	3.31
Dim=384, Depth=2, Heads=8, MLP=2.0, Conv=2, k=7, s=2	0.265	0.199	0.343	0.357	3.81
Dim=512, Depth=2, Heads=1, MLP=2.0, Conv=1, k=7, s=2	0.284	0.218	0.365	0.365	3.19
Dim=384, Depth=4, Heads=4, MLP=3.0, Conv=2, k=7, s=2	0.276	0.211	0.351	0.316	4.02

Table 4: Detailed mAP for small (S), medium (M) and large objects (L) using the Compact Convolutional Transformer ROI-head variants on VisDrone validation set.

We have mainly experimented with the hyperparameters the original CCT paper [HWS+21] suggests in their ablation studies. We notice that smaller configurations with 128 dimensions were relatively fast, running above 3 FPS on the GTX 1660, but they struggled to move past 0.25 mAP. Increasing the hidden dimension to 256 and depth to 6–7 layers improved accuracy to around 0.27–0.28 mAP, but these gains came at the expense of speed, in some cases dropping performance down to less than

2 FPS. Very large models with 768 dimensions or excessive depth plateaued at roughly 0.276–0.284 mAP while remaining computationally heavy, offering little benefit for the additional cost.

The most effective model we found was a mid-sized configuration with 384 dimensions, a 7×7 convolutional stem with stride 2, two convolutional layers, pooling with a 3×3 kernel and stride 2, and a transformer encoder with 4 layers, 4 heads, and an MLP ratio of 3.0. This setup achieved strong detection performance while maintaining practical efficiency, reaching competitive mAP levels and sustaining around 4 FPS. Considering both accuracy and throughput, this design proved to be the best CCTdeT variant, as it provided a measurable gain over the Faster R-CNN baseline and larger CCT heads, while still keeping inference speed close to real-time.

References

[CV18] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

[HWS+21] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *ArXiv*, abs/2104.05704, 2021.

[JQC23] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, January 2023.

[KH24] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *ArXiv*, abs/2410.17725, 2024.

[KTOW22] Junhyung Kang, Shahroz Tariq, Han Oh, and Simon S. Woo. A survey of deep learning-based object detection methods and datasets for overhead imagery. *IEEE Access*, 10:20118–20134, 2022.

[LDG+17] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[LGG+17] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2017.

[MGP23] Akhil Meethal, Eric Granger, and Marco Pedersoli. Cascaded zoom-in detector for high resolution aerial images. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2046–2055, 2023.

[NBN+25a] Mahya Nikouei, Bita Baroutian, Shahabedin Nabavi, Fateme Taraghi, Atefe Aghaei, Ayoob Sajedi, and Mohsen Ebrahimi Moghaddam. Small object detection: A comprehensive survey on challenges, techniques and real-world applications. *ArXiv*, abs/2503.20516, 2025.

[NBN+25b] Mahya Nikouei, Bita Baroutian, Shahabedin Nabavi, Fateme Taraghi, Atefe Aghaei, Ayoob Sajedi, and Mohsen Ebrahimi Moghaddam. Small object detection: A comprehensive survey on challenges, techniques and real-world applications. *ArXiv*, abs/2503.20516, 2025.

[NBSA24] Fnu Neha, Deepshikha Bhati, Deepak Kumar Shukla, and Md Amiruzzaman. From classical techniques to convolution-based models: A review of object detection algorithms. *2025 IEEE 6th International Conference on Image Processing, Applications and Systems (IPAS)*, CFP2540Z-ART:1–6, 2024.

[RF16] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2016.

[RHGS15] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[RKHD23] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. *ArXiv*, abs/2305.09972, 2023.

[SZJ+20] Pei Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse r-cnn: End-to-end object detection with learnable proposals. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14449–14458, 2020.

[WLY+19] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. Cspnet: A new backbone that can enhance learning capability of cnn. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1571–1580, 2019.

[WSN22] Dwi Wahyudi, Indah Soesanti, and Hanung Adi Nugroho. Toward detection of small objects using deep learning methods: A review. In *2022 14th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 314–319, 2022.

[WXC+21] Haiping Wu, Bin Xiao, Noel C. F. Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31, 2021.

[WXL+21] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558, 2021.

[WXLS24] Shuo Wang, Chunlong Xia, Feng Lv, and Yifeng Shi. Rt-detrv3: Real-time end-to-end object detection with hierarchical dense positive supervision. *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1628–1636, 2024.

[YALZ21] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *ArXiv*, abs/2104.01318, 2021.

[ZCW+22] Wenqiang Zhang, Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Qian Zhang, and Wenyu Liu. Featurized query r-cnn. *ArXiv*, abs/2206.06258, 2022.

[ZWB+18] Peng Fei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *ArXiv*, abs/1804.07437, 2018.

[ZWR+20] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 52:8574–8586, 2020.