

Article

ACT-FRCNN: Progress Towards Transformer-Based Object Detection

Sukana Zulfqar ¹, Zenab Elgamal ^{2,*} , Muhammad Azam Zia ¹, Abdul Razzaq ³ , Sami Ullah ⁴ and Hussain Dawood ² 

¹ Department of Computer Science, University of Agriculture Faisalabad, Faisalabad 38000, Pakistan; sukanazulfqar5517@gmail.com (S.Z.); mazamzia@uaf.edu.pk (M.A.Z.)

² School of Computing, Skyline University College, Sharjah P.O. Box 1797, United Arab Emirates; dawood.hussain@skylineuniversity.ac.ae

³ Department of Computer Science & IT, Muhammad Nawaz Sharif University of Agriculture Multan, Multan 59300, Pakistan; abdul.razzaq@mnsuam.edu.pk

⁴ Department of Computer Science, Government College University Faisalabad, Faisalabad 38000, Pakistan; samiullah@gcuf.edu.pk

* Correspondence: zenab.elgamal@skylineuniversity.ac.ae

Abstract: Maintaining a high input resolution is crucial for more complex tasks like detection or segmentation to ensure that models can adequately identify and reflect fine details in the output. This study aims to reduce the computation costs associated with high-resolution input by using a variant of transformer, known as the Adaptive Clustering Transformer (ACT). The proposed model is named ACT-FRCNN. Which integrates ACT with a Faster Region-Based Convolution Neural Network (FRCNN) for a detection task head. In this paper, we proposed a method to improve the detection framework, resulting in better performance for out-of-domain images, improved object identification, and reduced dependence on non-maximum suppression. The ACT-FRCNN represents a significant step in the application of transformer models to challenging visual tasks like object detection, laying the foundation for future work using transformer models. The performance of ACT-FRCNN was evaluated on a variety of well-known datasets including BSDS500, NYUDv2, and COCO. The results indicate that ACT-FRCNN reduces over-detection errors and improves the detection of large objects. The findings from this research have practical implications for object detection and other computer vision tasks.



Citation: Zulfqar, S.; Elgamal, Z.; Zia, M.A.; Razzaq, A.; Ullah, S.; Dawood, H. ACT-FRCNN: Progress Towards Transformer-Based Object Detection.

Algorithms **2024**, *17*, 475. <https://doi.org/10.3390/a17110475>

Academic Editor: Frank Werner

Received: 5 July 2024

Revised: 30 August 2024

Accepted: 31 August 2024

Published: 23 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: adaptive clustering transformer (ACT); object detection; segmentation; computer vision; classification; feature map

1. Introduction

The Bidirectional Transformer [1] paradigm has demonstrated outstanding improvements in machine translation [2], question–answering [3], text categorization [4], and document summarization [5], among other NLP issues. This accomplishment is partly due to the transformer’s ability to recognize complex relationships among input sequences through self-attention. It uses pre-train models on large datasets without encountering performance saturation due to its scalability [1,4].

Transformers can be directly applied to images by converting an image into a collection of patches. Although attention-based transformer models may not perform as well as convolution-based models on medium-sized datasets, these can be pre-trained on a large amount of data due to their capacity, as seen in NLP transformers [4]. This shows attention-based components have the potential to improve or even replace the traditional convolution approach that has been standard in vision modeling for a long time [5].

The higher-level feature maps of a convolutional network often align with the input’s spatial layout. This means that features from objects on the left of an input image tend

to appear on the left of higher-level feature maps [5]. This property makes it easy to convert convolutional classifiers into object detectors by adding detection heads that output object classes and positions on high-level convolutional maps. Unlike convolutional networks, transformers can work at each layer of the network, potentially affecting the spatial relationship between input features and those at the middle layers. This raises the question of whether ACT can be adjusted to perform tasks that require more sensitivity to local context, such as object detection or segmentation.

This research proposes an ACT technique to replace the transformer in the encoder of the original FRCNN system as an easy-to-add component. The ACT works well with the classic transformer and does not require new skills. Adding Multi-Task Knowledge Distillation (MTKD) to ACT can improve its accuracy to match the original transformer. MTKD can also allow for a seamless transition between models with different computing resources and accuracy during inference. However, some transformer adjustments can be made to reduce computational complexity [6–9]. If we have enough processing power, one obvious question is if we can improve the performance of FRCNN and strike a balance with computational resources.

In this paper a new model ACT-FRCNN is developed. The model is based on ACT and includes specific task heads for detecting and locating objects in images. The aim is to demonstrate that ACT-FRCNN, with its transformer-based backbone, can effectively preserve crucial spatial information for object detection. The experiment illustrates that ACT-FRCNN shares many of the desired features of transformer-based models while achieving strong performance on various benchmarked datasets including: BSDS500 [10], NC2016 [11], and COCO 2017 [12]. This research examines the benefits of using the extensive pre-training method with transformers for object detection tasks. The experimental results demonstrate a decrease in false positive detections and improved detection performance for large objects, possibly due to the architecture's ability to attend globally.

In this paper a large-scale computer vision models are typically trained on massive datasets and then fine-tuned for specific tasks.

The main contributions are threefold:

- Designed an ACT-FRCNN model that can be used for object detection and segmentation.
- Proposed an Adaptive Clustering Transformer (ACT) with Convolutional Neural Network (CNN) as the backbone of ACT-FRCNN.
- ACT-FRCNN is evaluated on three different datasets: BSDS500, NYUDv2, and COCO 2017, and demonstrate significant outcomes.

The remaining manuscript sections are organized: Section 2 discusses related work. Section 3 demonstrates the method and materials used in this research. Section 4 describes the experimental procedure. Finally, Sections 5 and 6 presents the discussion and conclusion respectively.

2. Related Work

2.1. NLP and CV for Attention Model

The attention model [13] has been widely utilized in computer vision (CV) and NLP due to its inherent capability to gather evolving information that evolves over time. The transformer facilitates the exchange of data between two entities, such as regions in CV or words in NLP. Transformer has achieved cutting-edge results in Machine translation [14], object detection [15], multimodality reasoning [16], picture classifications [17], video classifications [18], and language understanding [1]. The transformer has performed very well in different situations.

However, scaling is complex because the length of the input sequence leads to quadratic complexity. There are many ideas for solving the Transformer's computational issues. Kitaev et al. [19] suggested sharing the key and query, clustering features that are close together into one cluster using Locality Sensitive Hashing (LSH) to perform information exchange into each cluster. Zhu et al. [20] approximates the SoftMax between the key and query interaction with linear complexity, verifiable approximation error, and positive

orthogonal random features (PORFs). Using the association property, the authors of linear attention [9] reduce the complexity of key-query value multiplication from quadratic to linear. By eliminating the input of one layer at a time, the method of progressive elimination [8] can reduce the computational cost of the transformer while still achieving the same or better performance as the original. To reduce the computational complexity, asymmetric attention [21] summarizes essential features into a small number of key vectors by pooling over these features at many scales. Global graph reasoning [5] uses weighted pooling to compress input data into globally unique vectors that may then be transmitted. As mentioned earlier, these techniques altered the original transformer's structure and required substantial training and inference resources. Additionally, the performance gap between ACT and the original transformer can be further reduced with a few iterations of fine-tuning knowledge distillation [22].

2.2. Deep Learning for Object Detection

The framework for object detection sorts data within a changing time window [22]. The Viola-Jones Sliding window technique was used for face detection with a face detector [23] and AdaBoost [24]. In addition, the CNN method has been used for object classification with great success [25], as will the deep features classification used for object identification. The object detection utilizing in-depth features can be classified into two types: one-stage and two-stage approaches. The two-stage solutions include RCNN [26], Fast RCNN, and Faster RCNN [26], Whereas the one-stage options are YOLO [26] and SSD [27]. Training and deploying earlier techniques for object detection is more difficult due to the extensive post-processing pipeline NMS [28], imbalanced loss, and hand-crafted anchor [29]. Instead of using a sliding window, objects can now be detected by solving a robust set prediction problem against permutations. Liu et al. [30] developed an end-to-end people detection using the CNN encoding of visual features and the LSTM [31] decoding of the bounding box sequentially. Hungarian loss [31] will be applied to the end-to-end training process to ensure the predicted bounding box is consistent with the ground truth. Recurrent instance segmentation [32] superimposes an extra-segmentation head over the end-to-end object identification architecture, testing the concept of instance segmentation with great success. Using the more powerful transformer instead of LSTM [31] has simplified the object detection pipeline by solving the end-to-end set prediction problem, removing the need for custom priors. However, because self-attention has quadratic complexity, end-to-end set prediction converges slowly and requires high inference costs. SMCA [33] increased convergence speed by introducing Gaussian before the transformer decoder.

2.3. Self-Attention for Object Detection

DETR [5] is an essential method because it effectively utilizes transformers for object detection. It includes a set-matching loss, a transformer encoder, and a decoder, all integrated with a standard convolutional neural network (CNN) model such as ResNet-50/101. This method stands out because it eliminates the need for non-maximum suppression (NMS), and its decoder design can independently learn to remove redundant bounding box predictions. Deformable DETR [27] addresses a few drawbacks of DETR: (1) achieving relatively poor detection performance on small objects and (2) requiring substantially longer training epochs for convergence than standard detectors. Our proposed ACT aims to reduce the computational cost of DETR's inference without retraining, unlike Deformable DETR and SMCA.

DETR variations have also been suggested as solutions to the issues. By combining a multi-scale deformable encoder with learnable sparse sampling, Deformable DETR [34] speeds up convergence. To address problems mostly created by the insurmountable challenge of processing high-resolution feature maps, deformable DETR implements a deformable attention module that learns to focus on a small fraction of sample points in the feature map. Differently, RelationNet++ [35] benefits from multi-head attention. A "Bridging Visual Representations" (BVR) module built on a transformer decoder is

suggested to integrate information from various object representations. These results build on the work conducted by RelationNet [36] and Non-Local Networks (NL) [19] in object recognition by splitting the focus between bounding box features and pixel features.

DETR and Deformable DETR use convolutional networks to encode visual characteristics, while the transformer decodes these features into detection outputs. This article explores a new variation of a transformer-based detector in which the transformer encodes visual information instead of a standard region proposal network (RPN) and also generates the output detections.

2.4. Pertaining Model and Self-Training

Numerous studies have used large-scale datasets, like JFT-300M [37] and IG-940M [38], to investigate how useful pre-training is for learning visual representations. Big Transfer (BiT) [39] found that detection transfer performance was aided by extensive classification-based pre-training. He et al. [40] have found that when the detection dataset is large enough, small-scale classification-based pre-training, such as pre-training on ILSVRC-2012 [41], does not always improve detection performance compared to training from scratch. In this paper, the pre-training paradigms for object detection are the primary focus of our attention. However, semi-supervised learning [42] and self-training [43] paradigms are also effective when working with enormous amounts of unlabeled data.

2.5. Research Gap

The research discussed in this paper aims to improve our understanding of how transformer models, specifically the ACT model, can be used in object identification and other computer vision tasks. It also aims to enhance object detection systems by addressing the limitations of current methods. The paper proposes an FRCNN model that integrates ACT, leveraging the well-known transformer features of extensive pretraining capability and fast fine-tuning performance. Experimental results demonstrate the superior performance of the ACT-FRCNN model on the BSDS500, NYUDv2, and COCO 2017 benchmark datasets. The findings indicate that the ACT-FRCNN model outperforms many state-of-the-art techniques, achieving higher accuracy using fewer parameters.

3. Method and Materials

In ACT-FRCNN, a detection network is used with Adaptive Clustering backbone. This addition allows us to determine bounding box classifications and coordinates. The original transformer's design is identical to the adaptive clustering transformer (ACT) we proposed. ACT reduces the calculation cost of the transformer without requiring retraining. Through a few rounds of knowledge distillation fine-tuning, we can narrow the performance gap between ACT and the base transformer [1]. Figure 1 shows a visual representation of the ACT-FRCNN architecture. The RPN generates multiple predictions for each location on the map to identify exciting areas likely to contain objects. Each prediction corresponds to a specific anchor on the map and has different sizes and aspect ratios. The first step is to determine if an object exists in that area, and then the system determines the coordinates of a box around the object. Parameterization allows us to calculate the bounding boxes as distances from the anchor boxes.

$$\begin{aligned}
 t_x &= (x - x_a)/w_a & t_x^* &= (x^* - x_a)/w_a \\
 t_y &= (y - y_a)/h_a & t_y^* &= (y^* - y_a)/h_a \\
 t_w &= \log(w/w_a) & t_w^* &= \log(w^*/w_a) \\
 t_h &= \log(h/h_a) & t_h^* &= \log(h^*/h_a)
 \end{aligned} \tag{1}$$

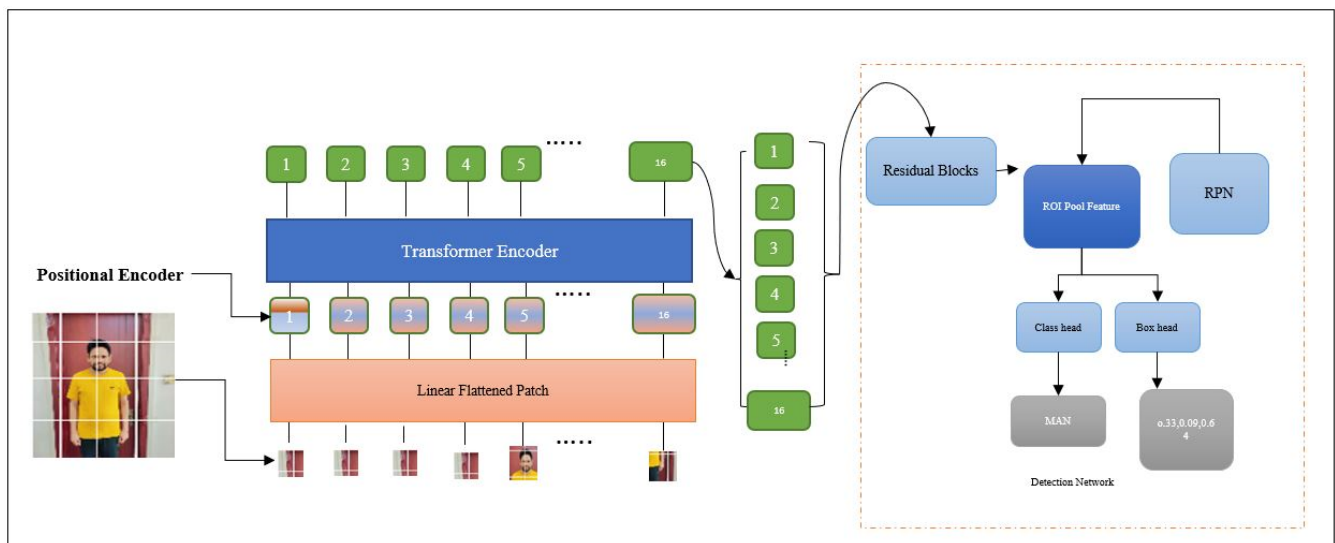


Figure 1. The ACT-FRCNN model relies on an ACT framework for its object detection capabilities. It generates a feature map using the last transformer layer’s per-patch outputs to feed into a detection model. For precise object detection, this model generates both class predictions and box coordinates. ACT-FRCNN exhibits remarkable performance and shares the benefits of transformer-based models.

The coordinates (x, y, w, h) of the box’s center, width, and height are denoted by x , x_a , and x^* , whereas x , x_a , and x^* represent the prediction, anchor, and ground truth, respectively. When comparing the true offsets t^* to the projected offsets t , the box training loss is equivalent to a Huber loss. The best candidates are used to RoI-pool regions from the feature map based on the RPN’s thorough forecast of object-containing regions, yielding one feature proposal for each region. The same parameterization as before provides these pooled features to a pair of lightweight heads, which then generate bounding box regressions and predictions for object categories (or background).

The DETR encoder includes features that allow it to use the attention mechanism to gather information from nearby positions. Our research shows that similar features in close physical proximity tend to produce identical attention maps. ACT must select prototypes representing the whole and communicate updated characteristics to its neighbor since its self-attention overlaps with its nearest neighbor. The more complex the encoder becomes, the more similar its characteristics grow as they start to share data. Also, each encoder layer shows a significant variation in the distributions of input features. These findings suggest using an adaptive method to determine the required number of prototypes for each data layer instead of relying on a predetermined set of cluster centers.

To tackle the issue of encoder attention redundancy, ACT utilizes key-query attention for representative prototypes. These prototypes are created by clustering similar query features and then averaging them. The modified features of the prototypes are then shared with their neighbors by using the Euclidean distance on the query feature space. Due to the wide variety of encoder characteristics, we need an adaptive clustering method to categorize features based on their location in the input and encoder layers. Our technique called multi-round exact Euclidean locality sensitivity hashing (E2LSH) [19] achieves query feature distribution-aware clustering.

3.1. Implementation Details

In this implementation, the used settings are similar to those for ACT-FRCNN and the original Faster R-CNN. The RPN predicts 15 anchors at each spot: the used of five sizes (32×32 , 64×64 , 128×128 , 256×256 , and 512×512 pixels) and three different shapes (1:1, 1:2, and 2:1). NMS is applied to the region suggestions generated by the RPN with an overlap threshold of 0.6. During training, the top 22,000 suggestions are sent to the detector

heads, and during inference, the top 800 are used. To create our final detections, the perform a final round of NMS at the time of inference using a threshold of 0.4. to explored two ways to use the encoder outputs: combining all intermediate outputs into a single one and using only the last one. The use a learned transformation in both approaches to reduce the spatial feature map's dimensionality to 512 channels. Finally, added intermediate residual blocks between the detection module and the encoder spatial feature map was practical (for a visual representation of the residual block structure, similar to ResNet with some slight modifications, refer to Figure 2).

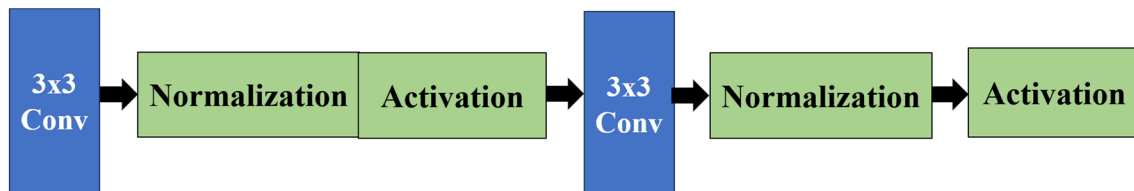


Figure 2. The encoder outputs are connected to the detection module using the intermediary residual convolution blocks. Batch normalization and GeLU activation functions are used in all studies.

3.2. Adaptive Clustering Transformer

3.2.1. Determine Prototypes

The locality sensitive hashing (LSH) prototype is used to join queries that are close together based on Euclidean distance. This method effectively solves the nearest neighbor search problem. When vectors are close, they are likely to have the same hash, while those far apart do not. By adjusting the hash function's parameters and the number of hashing iterations, to ensured that all vectors with a distance less than p end up in the same hash bucket. Specifically, the used the exact Euclidean locality sensitive hashing (E2LSH) as our hash function [19].

$$h(\vec{v}) = \left\lceil \frac{\vec{a} \cdot \vec{v} + b}{r} \right\rceil \quad (2)$$

where $h : R^d \rightarrow Z$ is the hash function, r is a hyper-parameter, and $\vec{a}, b$ are random variables satisfying $\vec{a} = (a_1, a_2, \dots, a_d)$, with $a_i \sim N(0, 1)$ and $b \sim u(0, r)$. The results carry more weight when LSH is applied iteratively. The final hash value obtained by

$$h(\vec{v}) = \sum_{i=0}^{L-1} B^i h_i(\vec{v}) \quad (3)$$

where each h_i is calculated using Equation (3) with parameters $\vec{a}$ and b drawn at random, and where B is a constant equal to 4 in our trials. The random normal vectors $\vec{a}_1, \vec{a}_2, \dots, \vec{a}_L$ and offset b_i in each hash function h_i can be visualized as a series of parallel hyperplanes. The hyper-parameter r controls the spacing of the hyperplanes. More significant separation indicates greater importance. In addition, L hash functions partition the search area into a number of cells, each of which contains vectors that are hashed to the same value. The likelihood that the vectors enter the same cell increases as the Euclidean distance decreases.

To obtain the prototypes, the hash value for each query was first computed. The questions that have the same hash value will be clustered together, with the prototype query serving as the group's hub. The $Q \in R^{N \times D_k}$ and the queries $P \in R^{C \times D_k}$ sets as prototypes, where N and C are the number of groups for queries and prototypes, respectively. The G_i represent the index of the cluster to which Q_i belongs. The Prototype of the j th group can be obtained by Equation (4).

$$P_j = \frac{\sum_i G_i = j Q_i}{\sum_i G_i = j 1} \quad (4)$$

3.2.2. Estimate Attention Output

After the initial stage, a prototype created to represent each set of questions. Then compute the attention map between the prototypes and keys. After that, the gathered of the specific target vector for each prototype and applied it to the original questions. This allows to estimate the impact on attention roughly. By reducing the number of prototypes and hash rounds, which are much fewer than N and M , this significantly reduce the computation's difficulty by a factor of C compared to the exact attention calculation.

Formally, the values $K \in R^{M \times D_k}$ are the keys $V \in R^{C \times D_k}$. The following equations provide us with an approximation of the output of attention:

$$L = L_{pred}(Y, Y_2) + L_{KD}(B_1, B_2) \quad (5)$$

$$\tilde{W} = \tilde{A}V \quad (6)$$

$$\tilde{V}^o = \tilde{W}_j, G_i = j \quad (7)$$

where G_i stands for the index of the cluster Q_i . SoftMax function is applied row-wise.

3.3. Multi-Task Knowledge Distillation

ACT has been found to reduce the computational cost of FRCNN without need of retraining. Additionally, it has been discovered that fine-tuning MTKD multiple times, leads to better performance-to-computability ratio. The process for MTKD is shown in Figure 3. Image features will first be extracted using the pre-trained CNN backbone, which is specifically designed to capture intricate spatial hierarchies and patterns within the data. Once extracted, these features will be passed into both the adaptive convolutional transformer (ACT) and the original transformer in parallel, allowing for a comparative analysis of their respective outputs. To ensure a seamless transition and integration between ACT and the original transformer, the training process will be guided by Multi-Task Knowledge Distillation (MTKD). This approach facilitates optimal performance by leveraging the strengths of both models. Below is a notation for the training loss:

$$L = L_{pred}(Y, Y_2) + L_{KD}(B_1, B_2.detach()) \quad (8)$$

where Y stands for the real world, B_1 for the ACT prediction's bounding box, and B_2, Y_2 for the whole prediction. The difference between the truth and FRCNN's prediction is denoted by $L_{pred}(Y, Y_2)$. The L_2 distance between the ACT and FRCNN predictions is minimized by $L_{KD}(B_1, B_2)$, which is the knowledge distillation loss.

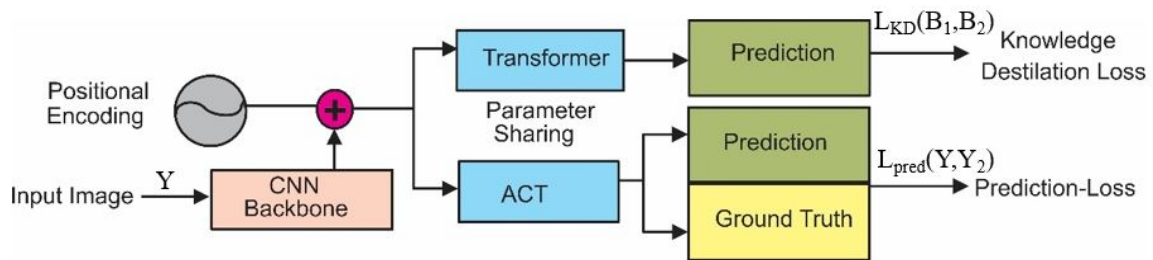


Figure 3. Multi-task knowledge Distillation. Initially, the CNN core extract features from images. The transformer kept in parallel with the excised section being fed into it. Training process guided by MTKD to ensure a smooth transition from the original transformer to ACT.

The training loss helps with the smooth transition from ACT to transformer by training the original transformer at the same time. The knowledge transformer includes region categorization and regression distillation features. The impact of ACT's approximation error is more noticeable in the regression sub-branch than in the classification sub-branch. The transferred of knowledge of the bounding box regression technique. Just by moving the box regression branch, the convergence rate is significantly improved.

3.4. Datasets

The ACT-FRCNN model is evaluated on the BSDS500 [10], NYUDv2 [11], and COCO 2017 [12] benchmarks to determine how well the suggested strategy performed in different settings. The studies conducted on the BSDS500, NYUDv2, and COCO 2017 benchmarks provide strong evidence for the effectiveness and adaptability of the proposed strategy and demonstrate how it can advance the field of computer vision. The aimed is to illustrate the efficiency and robustness of the proposed method in various image analysis tasks by evaluating it across these three benchmarks. According to the experimental results, this strategy outperforms the state of the art regarding accuracy and efficiency across all three benchmarks. These results showcase the versatility of the technique, indicating its suitability for tasks such as object identification and segmentation, depth estimation, and image segmentation.

3.4.1. BSDS500 Dataset

The Berkeley Segmentation Dataset and Benchmark dataset (BSDS500) comprises 500 natural RGB photos, with 200 allocated for training, 100 for validation, and 200 for testing. Each photograph is meticulously annotated by an average of five different individuals. The ACT-FRCNN model is tested on the evaluation set after being trained on the training and validation sets. To augment the dataset, each image can be flipped and rotated, increasing the dataset size 16-fold. On the other hand, the PASCAL VOC Context Dataset [44], which contains full-scene segmentation annotations with more than 400 classes and comprises 10,103 training images, used as supplementary training data as done in earlier research [45]. In Stage One of model training, the outer borders from the segmentation annotations are used to infer context and semantic signals. Therefore, Stage One is pre-trained using the PASCAL VOC Context Dataset [44] before refining it using the BSDS500 [10]. Stage One training is the only phase that utilizes the PASCAL VOC Context Dataset [44], which offers finer, more specific edge and boundary annotations.

3.4.2. NYUDv2 Dataset

The NYUDv2 dataset is widely used in computer vision research, especially for tasks such as depth estimation, scene understanding, and object detection. It comprises 1449 paired RGB and depth images, divided into 654 training images, 414 validation images, and 381 testing images. A key feature of NYUDv2 is its provision of dense depth maps, where each pixel in the image has a depth value, making it valuable for tasks like depth measurement and 3D reconstruction. The dataset also includes detailed ground truth annotations for object recognition and scene understanding, including scene category labels, object labels, and object borders. To augment the dataset, researchers often combine the training and validation sets and apply techniques such as four-fold rotation, random flipping, and scaling of both images and annotations [46].

3.4.3. COCO 2017 Dataset

The COCO 2017 Dataset experiments are performed using 118,000 training and 5000 validation images from the COCO 2017 detection and panoptic segmentation datasets [47]. The dataset includes bounding boxes and panoptic segmentation annotations for each image. On average, the training set contains seven instances per image, with object sizes varying from very small to large. Unless specified otherwise, the performance is evaluated using Average Precision (AP), measured with a range of Intersection over Union (IoU) thresholds. For the first 1000 photos in the COCO 2017 [12] validation set, the average Floating-point Operations per second (FLOPs) is also measured. The FLOPs computation includes operations involving convolutional layers, fully connected layers, attention matrix operations, and other components like clustering and heuristics.

4. Experimental Procedure

The experiments on ACT-FRCNN demonstrate its effectiveness on detection datasets BSDS500 [10], NYUDv2 [11], and COCO 2017 [12]. The experiments conducted on a system with two GPUs, each with 12 GB of memory, 64 GB of RAM and the Intel Core i9 12th generation processor. A thorough set of analyses on ACT-FRCNN is provided, showing how a transformer-based detector might be advantageous due to its simplified training process and ability to generalize well to unknown data. Object detection studies are conducted by torch vision [12] implementations of ResNet and Faster R-CNN (FRCNN). Object detection studies were conducted using the TorchVision [12] ResNet and Faster R-CNN (FRCNN).

4.1. Result Discussion

The Performance and Accuracy

The model's Average Precision (AP) is compared to DETR-DC5, with a specific focus on speed-accuracy trade-offs. Additionally, the adaptive clustering approach is compared to the K-means clustering method used in [46] where the number of clusters is denoted by C . When L and C are adjusted and the AP is calculated using various computational budgets, the ACT technique achieves significantly higher AP than K-means under an equalized computational budget. FLOPs are reduced from 168.9 GFLOPs to 164.3 GFLOPs compared to DETR-ACT, with only 0.5% loss in AP. MTKD substantially enhances AP, especially for devices with smaller FLOP capacities. ACT with $L = 32$ performs similarly to DETR-ACT, while ACT with $L = 16$ achieves a 4.6% increase in AP due to MTKD.

The advantages of proposed model over K-means clustering. K-means clustering requires the same number of clusters across all encoder levels. However, selecting an appropriate hyper-parameter can be difficult due to the varying distribution of queries based on the input image and encoder layer's index. Misusing the hyper-parameter C can result in inaccurate estimates or unnecessary computational costs. Furthermore, K-means may not converge properly in certain situations, and many clusters can empty if C is relatively large, as it is a hierarchical method. To address these issues, the proposed method employs an adaptive mechanism to dynamically calculate the number of prototypes.

Detailed AP Comparision: The Bounding Box Average Precisions (BBox APs) for the large (AP_L), medium (AP_M), and small (AP_S) instances are reported. Faster RCNN [48] and Discrete Event Time Series [15] is compared with these metrics. Both DETR and Faster R-CNN use Dilated ResNet-50 as the backbone. The results are presented in Table 1.

Table 1. A comparison of the AP of ACT-FRCNN model with DETR-ACT and Faster RCNN. Dilated ResNet-50 serves as the foundation of DETR-ACT and Faster RCNN. The '+' in the GFLOPs column denotes more flops in comparison to the backbone. The BBox APs for large, medium, and small instances are designated AP_L , AP_M , and AP_S , respectively.

Model	GFLOPs	AP	AP_L	AP_M	AP_S
Backbone (ResNet50-DC5)	110.7	---	---	---	---
DETR-DC5 [15]	+73.4	43.3	61.1	47.3	22.5
FasterRCNN-DC5 [49]	+209.3	41.1	55.0	45.9	22.9
DETR-ACT [50]	168.9	43.1	61.4	47.1	22.2
ACT ($L = 16$)	+47.2	41.4	60.8	43.1	17.1
ACT ($L = 20$)	+51.7	40.6	61.1	46.5	18.7
ACT ($L = 24$)	+55.2	39.2	61.3	47.3	20.2
ACT ($L = 32$)	+59.7	44.3	62.5	48.2	23.7
ACT + MTKD ($L = 16$)	+49.6	41.3	61.1	44.3	18.5
ACT + MTKD ($L = 20$)	+52.5	41.9	61.9	47.1	20.2
ACT + MTKD ($L = 24$)	+56.7	40.1	61.7	48.2	21.9
ACT + MTKD ($L = 32$)	+60.2	44.9	62.7	49.1	24.2

DETR-ACT performs better than our ACT model, with L equal to 16 while using fewer FLOPs. However, when it comes to detecting items of a certain size, ACT is much better than DETR-ACT. For large objects, our ACT model can provide a better approximation than FasterRCNN-DC5. The most significant AP losses occur with small- and medium-sized objects. Small- and medium-sized items experience the most AP losses. The most significant AP losses occur with small- and medium-sized objects. We suggest that our clustering attention can be considered as a dropout operation that can help prevent overfitting during the distillation of information.

Importantly, we found that the keys grouped using our strategy performed well. We tested query grouping, key grouping, and combining queries and keys. We adjusted the hyper-parameter L and compared the average precision on the validation set to ensure that the computational cost across all three experiments was consistent. Table 2 shows the results. Our model's ability to generalize is demonstrated by the fact that all three methods achieved similar average precision while using the same computational cost. This suggests that, when there are significantly more keys than queries, models can benefit from adaptively grouping the keys to improve performance.

Table 2. FLOPs and AP under various clustering targets.

Cluster Queries	Cluster Keys	FLOPs	AP
$L = 24$	$\times$	165.75	0.418
$\times$	$L = 18$	164.90	0.419
$L = 32$	$L = 12$	164.3	0.416

Inference time and memory: The above analyses were based on FLOPs or theoretical computation costs. We also performed field tests to assess the time and memory requirements. Table 3 shows the inference time and memory cost of the Nvidia GeForce GTX TITAN X for a batch size of 1. It is evident that ACT accelerates in real-world scenarios as predicted by the theoretical study, and it does so with significantly lower memory costs.

Table 3. The Nvidia GeForce GTX TITAN X's inference time and memory usage with a batch size of 1.

Model	Inference Time/Image	Memory
ACT ($L = 16$)	0.175 s	1010 MiB
ACT ($L = 20$)	0.190 s	1385 MiB
ACT ($L = 24$)	0.196 s	1500 MiB
ACT ($L = 32$)	0.201 s	1700 MiB
ACT-FRCNN	0.216 s	1850 MiB

4.2. Comparison with the State of Art

4.2.1. Results on COCO 2017 Dataset

The COCO 2017 dataset enables a comprehensive evaluation of detection models for domain shift. The models are assessed on the corresponding categories without any fine-tuning. The ViT-FRCNN models show significant improvement over the ResNet-FRCNN baselines on this dataset, as illustrated in Table 4. ViT-B/16*-FRCNN achieves a 22.9 AP (a + 7 AP boost) compared to the ResNet101-FRCNN-FPN baseline. Larger pre-training datasets result in better out-of-distribution generalization (+1.8 AP boost). These findings suggest that pre-training detection models with large-scale transformers are promising for enhancing their effectiveness in realistic, high-stakes scenarios.

Table 4. ACT-FRCNN outperforms ResNet-FRCNN techniques in terms of out-of-domain performance as measured by average precision (AP) in the COCO dataset.

Model	AP	AP ₅₀	AP ₇₅	AP _L
ResNet101-FRCNN-FPN	15.9	34.1	12.9	16.1
ResNet50-FRCNN-FPN	13.9	30.8	10.2	14.1
ViT-B/32*-FRCNN	17.9	36.4	15.8	18.0
ViT-B/32-FRCNN	16.4	34.5	14.0	16.5
ViT-B/16*-FRCNN	22.9	46.6	20.9	23.0
ViT-B/16-FRCNN	20.2	40.5	17.8	20.4
ACT-FRCNN	24.7	48.5	22.6	25.1

4.2.2. BSDS500 Dataset

The model's efficacy is measured against both conventional detectors, such as SCG [51], Sketch Tokens [52], SE [53], and MES [54], and deep learning-based detectors, such as Deep Boundary [55], COB [56], AMHNet [57], CED [58], LPCB [59], BDCN [50], DexiNed [60], PiDiNet [61], and EDTER-MS-VOC [62]. Table 5 displays the numerical outcomes. Our method is superior to all current state-of-the-art edge detectors, with accuracy levels of 86.5% (ODS), 88.3% (OIS), and 92.4% (AP).

Table 5. Outcomes on the BSDS500 dataset.

Method	Year	ODS	OIS	AP
SCG [46]	2012	0.739	0.758	0.773
Sketch Tokens [63]	2013	0.272	0.746	0.780
SE [64]	2014	0.746	0.767	0.803
MES [48]	2015	0.756	0.776	0.756
DeepBoundary [65]	2015	0.813	0.831	0.866
CED [51]	2017	0.815	0.833	0.889
LPCB [52]	2018	0.815	0.834	-----
BDCN [45]	2019	0.828	0.844	0.890
DexiNed [53]	2020	0.729	0.745	0.583
PiDiNet [54]	2021	0.807	0.823	-----
EDTER-MS-VOC [55]	2022	0.848	0.865	0.903
ACT-FRCNN (our)	-----	0.865	0.883	0.924

4.2.3. Results on NYUDv2 Dataset

In NYUDv2, we conducted tests on RGB images and evaluated them in comparison to cutting-edge techniques, such as gPb-ucm [10], Silberman [11], gPb+NG, SE [53], SE+NG+OEF, Semi Contour, HED, RCF [46], AMH-Net [57], LPCB [59], BDCN [50], PiDiNet [61], and EDTER [61]. On a single-scale input, all outcomes are based. The numeric outcomes of our approach and those of our rivals are displayed in Table 6. Our approach results in the top ODS, OIS, and AP scores of 81.3%, 82.0%, and 85.1%, respectively. We improved the results in three categories by 3.9%, 3.1%, and 3.2% when compared to the second-best method.

Table 6. Quantitative comparisons on NYUDv2.

Method	Year	ODS	OIS	AP
gPb-ucm [10]	2011	0.632	0.661	0.562
Silberman [11]	2012	0.658	0.661	-
gPb+NG [56]	2013	0.687	0.716	0.629

Table 6. Cont.

Method	Year	ODS	OIS	AP
SE [64]	2014	0.695	0.708	0.679
SE+NG+ [57]	2014	0.706	0.734	0.738
OEF [58]	2015	0.651	0.667	-
Semi Contour [59]	2016	0.680	0.700	0.690
HED [60]	2015	0.720	0.734	0.734
RCF [61]	2017	0.729	0.742	-
AMH-Net [62]	2017	0.744	0.758	0.765
LPCB [52]	2018	0.739	0.754	-
BDCN [45]	2019	0.748	0.763	0.790
PiDiNet [54]	2021	0.733	0.747	-
EDTER [55]	2022	0.774	0.789	0.797
ACT-FRCNN (Our)	-----	0.813	0.820	0.831

5. Discussion

The developments in object detection have received numerous inputs from different models and algorithms, especially the ACT-FRCNN, which is based on transformers. The results of the experiments on datasets, such as BSDS500, NYUDv2, and COCO, clearly show that our proposed ACT-FRCNN achieved a better performance than the traditional models. Thus, incorporating ACT with the detection task head has helped improve the effectiveness and proficiency in detecting objects within images.

It draws attention to the necessity of an advanced architecture to be incorporated with the increasing efficacy of the object detection task. Using the conditional generative adversarial network utilized in developing the current model further enhances the detection features by creating realistic samples to improve the training procedure. Notably, this architecture of transformers and generative models not only enhances the accuracy of detections but also makes the results invariant to the context and other conditions.

The research has benefited from collaboration between institutions: the University of Agriculture Faisalabad, Skyline University College, Sharjah-UAE, Muhammad Nawaz Sharif University of Agriculture Multan, and Government College University Faisalabad. The collaboration with other institutions, especially in organizations, has helped accrue a wide variety of insights and expertise, thus contributing to the complete development and testing of the ACT-FRCNN mode.

Consequently, our work demonstrates that transformer models have a promising future for object detection tasks. The findings of multiple experiments demonstrate that, even when trained with law-breaking samples, the ACT-FRCNN model does even better than the current state of the art. More research should be carried out on applying such new architectures to further advance the possibilities of object detection.

6. Conclusions

An object detection solution based on ACT-FRCNN, which demonstrates robust performance through a transformer backbone, is presented in the proposed research. As described in these findings, alternative architectures, like transformers, are more promising than adaptive clustering transformers (ACTs) for advancing complex vision tasks. Transformer-based models have demonstrated impressive capabilities in pre-training on large datasets without reaching saturation. ACT-FRCNN demonstrates the advantages of using transformers in vision tasks. For instance, in the NYUDv2 dataset, ACT-FRCNN achieved ODS, OIS, and AP scores of 81.3%, 82.0%, and 85.1%, respectively, outperforming the previous best methods by 3.9%, 3.1%, and 3.2%, respectively. These results highlight the advantages of transformer-based models, specifically ACT, in vision tasks. This marks an important milestone, as it is the first exploration of transformer-based architectures for solving various vision problems in the research community. As a result of the success of ACT-FRCNN, further research and development in computer vision using transformer-

based models is encouraged. Future research ideas might be built around novel transformer kinds and transformer topologies modified for object-detecting applications. Lastly, more advanced techniques seem promising to improve the results and performance rates of the particular method, such as Conditional Generative Adversarial Networks (CGANs), or conduct additional research into the suitability of transformers in combination with other machine learning frameworks, like federated learning.

Author Contributions: The study idea and its execution were contributed to by all authors. Methodology, formal analysis, project administration, data curation, and English language editing performed by S.Z., Z.E., M.A.Z., A.R., S.U. and H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: All procedures performed in studies were in accordance with the ethical standards of the institutional and/or national research committee.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Portes, J.; Trott, A.; Havens, S.; King, D.; Venigalla, A.; Nadeem, M.; Sardana, N.; Khudia, D.; Frankle, J. MosaicBERT: A bidirectional encoder optimized for fast pretraining. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 9–15 December 2024; Volume 36.
- Team, I. Internlm: A Multilingual Language Model with Progressively Enhanced Capabilities. 2023. Available online: <https://github.com/InternLM/InternLM> (accessed on 27 September 2023).
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 1–24.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
- Chen, D.; Cao, P.; Yan, L.; Chen, H.; Lin, J.; Li, X.; Yuan, L.; Wu, K. Multi-Scale Mixed Attention Tea Shoot Instance Segmentation Model. *Phyton* **2024**, *93*, 261.
- Chen, Y.; Rohrbach, M.; Yan, Z.; Shuicheng, Y.; Feng, J.; Kalantidis, Y. Graph-based global reasoning networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 433–442.
- Gao, P.; You, H.; Zhang, Z.; Wang, X.; Li, H. Multi-modality latent interaction network for visual question answering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5825–5835.
- Goyal, S.; Choudhury, A.R.; Raje, S.; Chakaravarthy, V.; Sabharwal, Y.; Verma, A. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 3690–3699.
- Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 5156–5165.
- Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 898–916.
- Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor segmentation and support inference from rgbd images. *ECCV* **2012**, 7576, 746–760.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13. Springer: Berlin/Heidelberg, Germany; pp. 740–755.
- Jiang, Z.; Gao, P.; Guo, C.; Zhang, Q.; Xiang, S.; Pan, C. Video object detection with locally-weighted deformable neighbors. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8529–8536. [[CrossRef](#)]
- Zhang, Y.; Liu, C.; Liu, M.; Liu, T.; Lin, H.; Huang, C.B.; Ning, L. Attention is all you need: Utilizing Attention in AI-enabled drug discovery. *Brief. Bioinform.* **2024**, *25*, bbad467.
- Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part I 16. Springer: Berlin/Heidelberg, Germany; pp. 213–229.

16. Gao, P.; Jiang, Z.; You, H.; Lu, P.; Hoi, S.C.; Wang, X.; Li, H. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, 15–20 June 2019; pp. 6639–6648.
17. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 10012–10022.
18. Wu, X.; Zhang, K.; Hu, Y.; He, X.; Gao, X. Multi-scale non-local attention network for image super-resolution. *Signal Process.* **2024**, *218*, 109362.
19. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The efficient transformer. *arXiv* **2020**, arXiv:2001.04451.
20. Choromanski, K.; Likhoshesterov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J.; Mohiuddin, A.; Kaiser, L.; et al. Rethinking attention with performers. *arXiv* **2020**, arXiv:2009.14794.
21. Zhu, Z.; Xu, M.; Bai, S.; Huang, T.; Bai, X. Asymmetric non-local neural networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 593–602.
22. Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; Ma, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3713–3722.
23. Husain, S.; Abolkasim, E. Estimating the Number of People in Digital Still Images Based on Viola-Jones Face Detection Algorithms. *Afr. J. Adv. Pure Appl. Sci. (AJAPAS)* **2024**, *3*, 146–154.
24. Emami, S.; Martínez-Muñoz, G. Sequential training of neural networks with gradient boosting. *IEEE Access* **2023**, *11*, 42738–42750. [[CrossRef](#)]
25. Pu, T.; Sun, M.; Wu, H.; Chen, T.; Tian, L.; Lin, L. Semantic representation and dependency learning for multi-label image recognition. *Neurocomputing* **2023**, *526*, 121–130. [[CrossRef](#)]
26. Li, J.; Zhu, Z.; Liu, H.; Su, Y.; Deng, L. Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R-CNN. *Ecol. Inform.* **2023**, *77*, 102210. [[CrossRef](#)]
27. Yang, F.; Huang, L.; Tan, X.; Yuan, Y. FasterNet-SSD: A small object detection method based on SSD model. *Signal Image Video Process.* **2024**, *18*, 173–180. [[CrossRef](#)]
28. Wang, L.; Zhang, X.; Zhao, F.; Wu, C.; Wang, Y.; Song, Z.; Yang, L.; Xu, B.; Li, J.; Ge, S.S. Fuzzy-nms: Improving 3D object detection with fuzzy classification in nms. *IEEE Trans. Intell. Veh.* **2024**; early access.
29. Wang, S.; Chen, R.; Wu, H.; Li, X.; Feng, Z. YOLOH: You Only Look One Hourglass for Real-time Object Detection. *IEEE Trans. Image Process.* **2024**, *33*, 2104–2115. [[CrossRef](#)]
30. Liu, Z.; Tan, Y.; Wu, W.; Tang, B. Dilated high-resolution network driven RGB-T multi-modal crowd counting. *Signal Process. Image Commun.* **2023**, *112*, 116915. [[CrossRef](#)]
31. Pandiyan, V.; Akeddar, M.; Prost, J.; Vorlaufer, G.; Varga, M.; Wasmer, K. Long short-term memory based semi-supervised encoder—Decoder for early prediction of failures in self-lubricating bearings. *Friction* **2023**, *11*, 109–124. [[CrossRef](#)]
32. Kong, S.; Fowlkes, C.C. Recurrent pixel embedding for instance grouping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9018–9028.
33. Gao, P.; Zheng, M.; Wang, X.; Dai, J.; Li, H. Fast convergence of detr with spatially modulated co-attention. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3621–3630.
34. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
35. Wu, H.; Lin, C.; Liu, J.; Song, Y.; Wen, Z.; Qin, J. Feature masking on non-overlapping regions for detecting dense cells in blood smear image. *IEEE Trans. Med. Imaging* **2023**, *42*, 1668–1680.
36. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
37. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 843–852.
38. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 181–196.
39. Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; Houlsby, N. Big transfer (bit): General visual representation learning. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part V 16. Springer: Berlin/Heidelberg, Germany; pp. 491–507.
40. He, K.; Girshick, R.; Dollár, P. Rethinking imagenet pre-training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4918–4927.
41. Chatterjee, A.; Mukherjee, J.; Das, P.P. ImageNet Classification using WordNet Hierarchy. *IEEE Trans. Artif. Intell.* **2023**, *5*, 1718–1727. [[CrossRef](#)]
42. Yalniz, I.Z.; Jégou, H.; Chen, K.; Paluri, M.; Mahajan, D. Billion-scale semi-supervised learning for image classification. *arXiv* **2019**, arXiv:1905.00546.

43. Ma, C.; Pan, X.; Ye, Q.; Tang, F.; Dong, W.; Xu, C. CrossRectify: Leveraging disagreement for semi-supervised object detection. *Pattern Recognit.* **2023**, *137*, 109280. [\[CrossRef\]](#)
44. Sarkar, H.; Chudasama, V.; Onoe, N.; Wasnik, P.; Balasubramanian, V.N. Open-Set Object Detection By Aligning Known Class Representations. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 1–6 January 2024; pp. 219–228.
45. He, J.; Zhang, S.; Yang, M.; Shan, Y.; Huang, T. Bi-directional cascade network for perceptual edge detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3828–3837.
46. Ren, X.; Bo, L. Discriminatively trained sparse code gradients for contour detection. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 3–6 December 2012; Volume 1, pp. 584–592.
47. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
48. Sironi, A.; Lepetit, V.; Fua, P. Projection onto the manifold of elongated structures for accurate extraction. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 316–324.
49. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
50. Zheng, M.; Gao, P.; Zhang, R.; Li, K.; Wang, X.; Li, H.; Dong, H. End-to-end object detection with adaptive clustering transformer. *arXiv* **2020**, arXiv:2011.09315.
51. Wang, Y.; Zhao, X.; Li, Y.; Huang, K. Deep crisp boundaries: From boundaries to higher-level tasks. *IEEE Trans. Image Process.* **2018**, *28*, 1285–1298. [\[CrossRef\]](#)
52. Deng, R.; Shen, C.; Liu, S.; Wang, H.; Liu, X. Learning to predict crisp boundaries. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 562–578.
53. Poma, X.S.; Riba, E.; Sappa, A. Dense extreme inception network: Towards a robust cnn model for edge detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1923–1932.
54. Su, Z.; Liu, W.; Yu, Z.; Hu, D.; Liao, Q.; Tian, Q.; Pietikäinen, M.; Liu, L. Pixel difference networks for efficient edge detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 5117–5127.
55. Pu, M.; Huang, Y.; Liu, Y.; Guan, Q.; Ling, H. Edter: Edge detection with transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1402–1412.
56. Gupta, S.; Arbelaez, P.; Malik, J. Perceptual organization and recognition of indoor scenes from RGB-D images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 564–571.
57. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part VII 13. Springer: Berlin/Heidelberg, Germany; pp. 345–360.
58. Hallman, S.; Fowlkes, C.C. Oriented edge forests for boundary detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1732–1740.
59. Zhang, Z.; Xing, F.; Shi, X.; Yang, L. Semicontour: A semi-supervised learning approach for contour detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 251–259.
60. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1395–1403.
61. Liu, Y.; Cheng, M.-M.; Hu, X.; Wang, K.; Bai, X. Richer convolutional features for edge detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3000–3009.
62. Xu, D.; Ouyang, W.; Alameda-Pineda, X.; Ricci, E.; Wang, X.; Sebe, N. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
63. Lim, J.J.; Zitnick, C.L.; Dollár, P. Sketch tokens: A learned mid-level representation for contour and object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 3158–3165.
64. Dollár, P.; Zitnick, C.L. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1558–1570. [\[CrossRef\]](#)
65. Kokkinos, I. Pushing the boundaries of boundary detection using deep learning. *arXiv* **2015**, arXiv:1511.07386.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.