Journal of Intelligent Learning Systems and Applications, 2024, 16, 175-221
https://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

Advancing Crowd Object Detection: A Review of YOLO, CNN and ViTs Hybrid Approach*

Mahmoud Atta Mohammed Ali*, Tarek Aly, Atef Tayh Raslan, Mervat Gheith, Essam A. Amin

Faculty of Graduate Studies for Statistical Research, Software Engineering Department, Cairo University, Giza, Egypt
Email: *mjksoft@gmail.com, tarekmmmmt@pg.cu.edu.eg, Dr.Atif.Raslan@gmail.com, mervat_ghelth@yahoo.com, essam.amin@cu.edu.eg

How to cite this paper: Ali, M.A.M., Aly, T., Raslan, A.T., Gheith, M. and Amin, E.A. (2024) Advancing Crowd Object Detection: A Review of YOLO, CNN and ViTs Hybrid Approach. *Journal of Intelligent Learning Systems and Applications*, **16**, 175-221.
https://doi.org/10.4236/jilsa.2024.163011

Received: June 15, 2024
Accepted: July 30, 2024
Published: August 2, 2024

Abstract

One of the most basic and difficult areas of computer vision and image understanding applications is still object detection. Deep neural network models and enhanced object representation have led to significant progress in object detection. This research investigates in greater detail how object detection has changed in the recent years in the deep learning age. We provide an overview of the literature on a range of cutting-edge object identification algorithms and the theoretical underpinnings of these techniques. Deep learning technologies are contributing to substantial innovations in the field of object detection. While Convolutional Neural Networks (CNN) have laid a solid foundation, new models such as You Only Look Once (YOLO) and Vision Transformers (ViTs) have expanded the possibilities even further by providing high accuracy and fast detection in a variety of settings. Even with these developments, integrating CNN, YOLO and ViTs, into a coherent framework still poses challenges with juggling computing demand, speed, and accuracy especially in dynamic contexts. Real-time processing in applications like surveillance and autonomous driving necessitates improvements that take use of each model type's advantages. The goal of this work is to provide an object detection system that maximizes detection speed and accuracy while decreasing processing requirements by integrating YOLO, CNN, and ViTs. Improving real-time detection performance in changing weather and light exposure circumstances, as well as detecting small or partially obscured objects in crowded cities, are among the goals. We provide a hybrid architecture which leverages CNN for robust feature extraction, YOLO for rapid detection, and ViTs for remarkable global context capture via self-attention techniques. Using an innovative training regimen that prioritizes flexible learning rates and data augmentation procedures, the model is trained on an extensive

*YOLO: You Only Look Once.
CNN: Conventional Neural Network.
ViTs: Vision Transformers.

dataset of urban settings. Compared to solo YOLO, CNN, or ViTs models, the suggested model exhibits an increase in detection accuracy. This improvement is especially noticeable in difficult situations such settings with high occlusion and low light. In addition, it attains a decrease in inference time in comparison to baseline models, allowing real-time object detection without performance loss. This work introduces a novel method of object identification that integrates CNN, YOLO and ViTs, in a synergistic way. The resultant framework extends the use of integrated deep learning models in practical applications while also setting a new standard for detection performance under a variety of conditions. Our research advances computer vision by providing a scalable and effective approach to object identification problems. Its possible uses include autonomous navigation, security, and other areas.

Keywords

Object Detection, Deep Learning, Computer Vision, YOLO, Convolutional Neural Networks (CNN), Vision Transformers, Neural Networks, Transfer Learning, Autonomous Driving, Self-Drive Vehicles

1. Introduction

One of the key applications of computer vision is object identification, which is essential to many other fields like robotics, autonomous driving, and surveillance. Convolutional neural networks (CNN) have emerged as the cornerstone of fashionable techniques to object detection, thanks to the major developments in deep learning technologies [1].

Nonetheless, the area is still developing, as evidenced by the introduction of new models like YOLO and Vision Transformers (ViTs) in recent times, which provide improved speed and accuracy of detection [2] [3].

However, there are still many obstacles to overcome before CNN, ViTs, and YOLO can all be combined into a single framework. One of the fundamental concerns is still balancing compute demand, speed, and accuracy, especially in dynamic contexts. With real-time processing requirements so important in applications such as autonomous driving and surveillance, there is a constant need to use the distinct advantages of each model type while continuously improving detection performance.

Even though more models were available at the time, all prior research, were restricted to providing an overview and comparison of a small number of object identification models.

The models were divided into two categories: two-stage and one-stage detectors in the earliest surveys using the same methodology. Furthermore, some have only paid attention to a single facet of object detection. One area of research, for instance, is the identification of conspicuous items. The detection of small items has been the subject of studies by others. They examine object de-

tecting models' learning techniques.

In order to overcome these obstacles, this study suggests an object detection system that integrates CNN, ViTs, and YOLO to maximize detection speed and accuracy while reducing processing requirements [4]. Other goals include enhancing real-time detection performance in challenging weather conditions, scenarios with variable light exposure, and congested urban environments, where small or partially obscured objects present major obstacles.

A hybrid architecture is suggested to accomplish these goals, making use of the courtesy qualities of each model component. Robust feature extraction is the responsibility of CNN, quick detection is the responsibility of YOLO, and global context is captured by ViTs using self-attention techniques (**Figure 1**). Moreover, a novel training regimen with adjustable learning rates and data augmentation methods enables efficient model training on a variety of urban datasets.

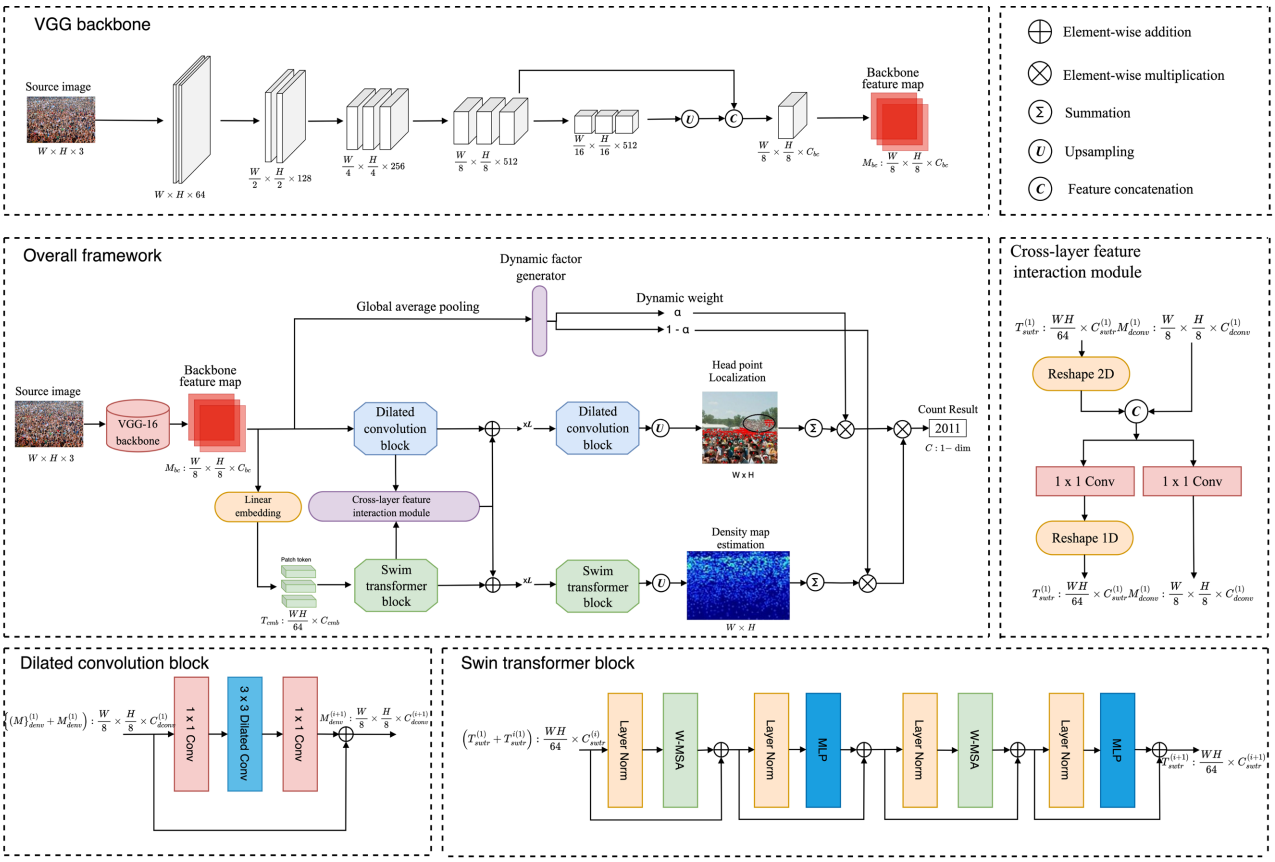

Figure 1. Detailed network architecture of the overall CLDE-Net[1] framework[2] [5].

Test results confirm the effectiveness of the suggested strategy. The combined model shows a notable increase in detection accuracy when compared to separate YOLO, CNN, or ViTs models, especially in difficult settings with severe occlusion and low light levels. Furthermore, the suggested model attains a note-

[1]Crowd Localization and Density Estimation [5].

[2]CLDE-Net: crowd localization and density estimation based on CNN and transformer network [5].

worthy reduction in inference time in contrast to baseline models, permitting real-time object detection without compromising performance.

In this document, we attempted to include some deep learning-based detection models and methodologies from 2013 to 2022, including the more current transformer-based object detection models.

The number of models we have included has not been thoroughly examined and analyzed in any other work. Additionally, we separated the detection models into four groups. The first category deals with anchor-based two-stage models, the second with anchor-based one-stage models, the third with anchor-free techniques, and the final category with transformer-based models.

This study presents a revolutionary approach to object identification that combines YOLO, CNN, and ViTs in a collaborative way to provide a comprehensive answer to object detection challenges. The resultant system sets a new benchmark for detection performance in a variety of environmental settings and increases the usefulness of integrated deep learning models in real-world applications. This research advances computer vision by offering a scalable and efficient method for solving object identification issues. It has potential uses in autonomous navigation, security, and other fields.

1.1. Problem Definition

The year 2001 marked a significant advancement in object detection and image recognition when Paul Viola and Michael Jones created an efficient facial detection system [6]. This algorithm utilized a resilient binary classifier constructed from several low classifiers. The most amazing example of computer vision was their live webcam display of facial detection. Navneet Dalal and Bill Triggs produced a new work in 2005. Their method performed better than previous pedestrian recognition algorithms and was based on the feature descriptor Oriented Gradient Histograms (HOG) [7]. Another important feature-based model, the Deformable Part Model (DPM) was created in 2009 by Felzenszwalb *et al.* [8].

Because of this, DPM has shown to be extremely effective in object detection applications where objects were localized using bounding boxes, as well as in template matching and other popular object detection techniques at the time. Numerous techniques for identifying items and extracting patterns from photos have already been developed [9] [10]. Traditional approaches typically consist of three components:

1) Using techniques like sliding window [11] [12], max-margin object recognition, and region proposal like the selective search algorithm [13], the initial stage entails examining the entire image at various scales and positions in order to create candidate boxes.

Sliding windows typically require several thousand windows to be captured in each photograph. Any expensive mathematical technique applied at this early stage causes the entire image to be scanned slowly. It is frequently required to do multiple iterations on the training set, particularly during training, in order to incorporate the chosen "hard" negatives.

2) To extract visual characteristics or image patterns, the second phase, feature extraction, analyzes the regions that are formed. Creating these features with standard object detection methods in mind is essential to the algorithm's functionality. We use techniques like Scale-Invariant Feature Transform (SIFT) [14], HOG [7], Haar-Like features [15], and Speeded Up Robust Feature to achieve this (SURF) [16], and BRIEF (Binary Robust Independent Elementary Features) [17].

3) The next stage is to classify these entities using techniques like Support Vector Machine (SVM) [18], Ad boost [19], Deformable Part-based Model (DPM) [15], and K-Nearest Neighbors [20], regardless of whether they contain an object or not. Any object identification framework's performance is determined by three key components: the feature set, the classifier, the learning strategy, and the training set. Specifically, most conventional techniques that have proven most effective in the recent PASCAL VOC detection challenges [21] have coupled numerous feature channels with detectors that have several aspects and mobile components (**Figure 2**).

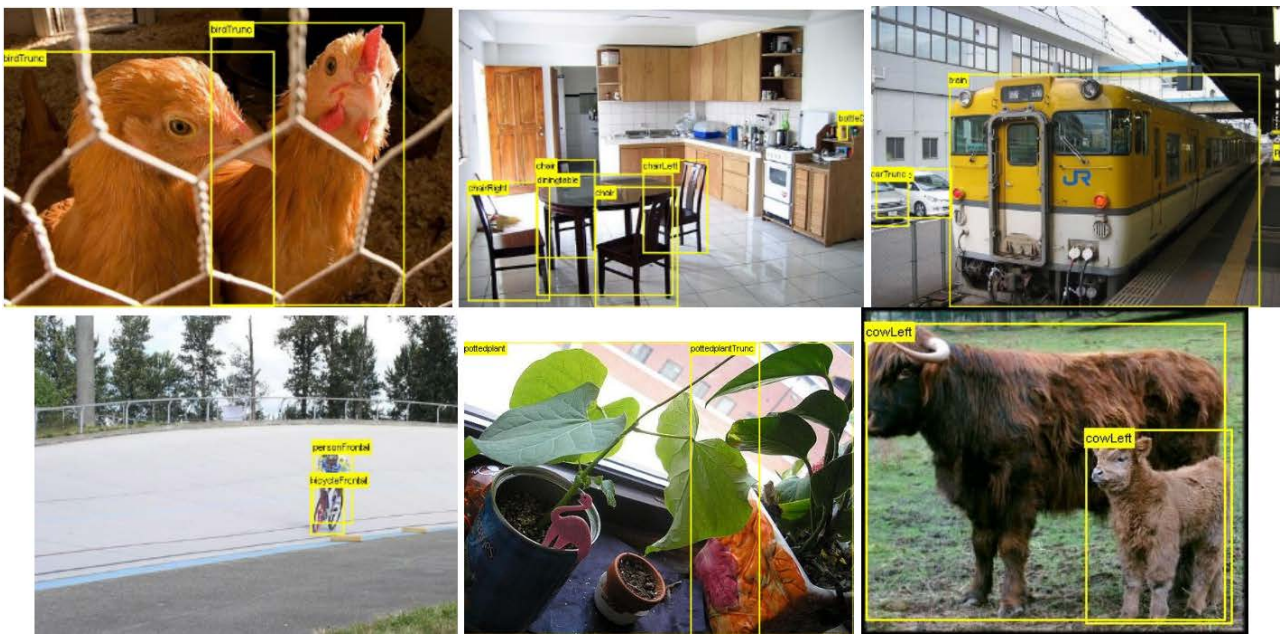

Figure 2. Samples from Pascal VOC 07 [22].

The PASCAL VOC trials carried out between 2008 and 2012 with these conventional methods had become mediocre, with only slight improvements; this has brought attention to the shortcomings of conventional detectors and the necessity of creating more reliable techniques. The problem with conventional methods, like the ones described above, that use sliding windows, for example, where a rectangle of different sizes moves across the entire image to find relevant objects, is that it takes a lot of computational work and creates many duplicate windows.

The subjacent classifier's effort has a crucial impact on the final product.

Conventional methods for object detection have relied on our ability to manually create features or models based on our understanding. To characterize and categorize filtered images, we try to look for patterns and edges. However, the most recent developments indicate that it is most effective to assign such jobs to the computer so that it can make its own discoveries.

In 2011, after the 2010 start of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [23], the competition's categorization error rate was roughly 26%. The error rate decreased to 16.4% in 2012 after a year thanks to the AlexNet convolution neural network model [3]. Its design is like that of Yann LeCun's LeNet-5 [24]. This made it a crucial turning point for convolutional neural networks at the time. Convolution neural networks have emerged victorious in the upcoming years and since 2012, resulting in a significant decrease in the classification error rate for ILSRVC.

In computer vision, crowd object detection is still a difficult issue because of the high object density, occlusion, different scales, and complicated backdrops. Although they work well in many situations, traditional object identification algorithms frequently have trouble maintaining high levels of efficiency and accuracy in congested environments. In order to overcome these obstacles, this research will create a hybrid approach that combines the best features of three cutting-edge approaches: Convolutional Neural Networks (CNN) for their potent feature extraction, Vision Transformers for their capacity to model global context, and YOLO for its real-time detection capabilities.

1.1.1. High-Density Object Detection

Owing to the abundance of things in proximity, crowded environments like public meetings, urban areas, and sporting events present a considerable obstacle. In such environments, current models frequently fall short in accurately detecting and differentiating objects [25] [26].

1.1.2. Occlusion and Overlapping Objects

Identifying individual objects in crowded settings is challenging for standard models because objects often overlap or occlude one another [27] [28].

1.1.3. Changing Scales and Complexity of Background

The detection procedure is further complicated by the fact that objects in crowded settings might appear at varying scales and against varied backdrops [29] [30].

1.1.4. Model Efficiency and Real-Time Processing

For real-world applications like autonomous navigation and surveillance, detection models that are both highly accurate and efficiently operate in real-time are required [3] [31].

1.2. Aims and Objectives

1.2.1. Aim

The main goal of this research is to create a hybrid model that combines CNN,

Vision Transformers, and YOLO to improve the efficiency and accuracy of crowd item recognition. The goal of this hybrid strategy is to enhance detection performance in high-density, occluded, and varied-scale situations by overcoming the shortcomings of individual models.

1.2.2. Objectives

1) Assess each detection model's performance

Task: Evaluate the individual performance of CNN, Vision Transformers, and YOLO in congested situations.

Method: Use individual datasets with packed sceneries and standard datasets such as COCO2017. Analyze performance indicators including average precision (AP), recall, and precision.

Expected Result: Determine each model's baseline performance in order to assess its advantages and disadvantages when managing busy scenarios.

2) Create a model for hybrid detection

Task: Create and put into practice a hybrid model that combines the robust feature extraction capabilities of CNN, the real-time detection capabilities of YOLO, and the global context modeling capabilities of Vision Transformers.

Method: Employ a common framework to integrate the models. To strike a compromise between accuracy and computing efficiency, make use of transfer learning strategies and architectural optimization.

Expected Result: Develop a strong hybrid model by utilizing CNN, Vision Transformers, and YOLO's complementary strengths.

3) Examine the proposed hybrid model in comparison to the most advanced models

Task: Compare the hybrid model with the most advanced object detection models currently in use.

Method: Use both custom and standard datasets for experiments. Make use of measures like processing time, average precision (AP), recall, and precision. To verify the results' significance, do statistical tests.

Expected Result: Show that the hybrid model performs more accurately and efficiently than both standalone models and the most recent state-of-the-art models.

4) Demonstrate how applicable it is in real-world situations

Task: is to validate the hybrid model in practical applications including crowd management, driverless cars, and surveillance systems.

Method: Implement the hybrid paradigm in both virtual and real-world settings. Keep an eye on how it performs in real-time circumstances and get input for future improvements.

Expected Result: Demonstrate the hybrid model's usefulness and efficacy in congested real-world settings, highlighting its potential for widespread implementation.

5) Detailed goals analysis

(A) Evaluation of selected models

a) YOLO:

i) Strengths: Quick and effective, appropriate for real-time applications.

ii) Weaknesses: In busy environments, there may be issues with occlusion and small objects.

iii) Evaluation metrics: include recall, AP, frame rate, and precision.

b) CNN:

i) Strengths: Outstanding at managing a range of scales and feature extraction**.**

ii) Weaknesses: Requires optimization for real-time performance; computationally intensive.

iii) Evaluation metrics: accuracy, recall, AP and feature extraction quality**.**

c) Vision Transformers:

i) Strengths: Able to represent global context and long-range dependencies.

ii) Weaknesses: Computer-intensive and maybe requiring huge datasets for training.

iii) Evaluation metrics: include memory, accuracy, contextual modelling ability, and AP.

(B) Creation of a hybrid model

a) Architecture Design:

i) To detect objects initially, integrate YOLO.

ii) Employ CNN to handle different scales and improve feature extraction.

iii) Use Vision Transformers to refine detections and grasp global context.

b) Enhancement Techniques:

i) Make use of pre-trained models by utilizing transfer learning.

ii) Use strategies such as quantization, trimming, and effective layer architecture to guarantee real-time performance.

c) Validation:

i) Test a lot on different kinds of data.

ii) Iteratively improve the model in response to real-world feedback and performance indicators.

(C) Comparative analysis

a) Benchmarking:

i) For consistent comparisons, use benchmark datasets (like COCO2017).

ii) Compare with cutting-edge models like as SSD, DETR, and Faster R-CNN.

b) Statistical Analysis:

i) To verify performance gains, use statistical tests (such as ANOVA and t-tests).

ii) Examine the trade-offs between computing efficiency and accuracy.

6) Real-world use:

(A) Deployment

a) Use the hybrid model in experimental programs, including self-driving cars or surveillance systems.

b) Track performance in real-time settings and collect user input.

(B) Scalability

a) Evaluate how well the model can be adjusted to various settings and circumstances.

b) Make that the model can process data in real time with a reasonable latency.

Clarity and direction for the project are ensured by this thorough analysis of the goals and objectives, which offers a comprehensive view of the study goals and the steps required to achieve them.

2. Literature Review

2.1. Introduction

A crucial field of study in computer vision is crowd object identification, which finds use in anything from driverless cars and crowd control systems to security and surveillance. Numerous techniques have been created over time to deal with the particular difficulties presented by crowded settings, such as high object density, occlusion, and different object scales. The objective of this literature review is to examine and evaluate the development of object identification methods, with a particular emphasis on the benefits and drawbacks of important strategies like You Only Look Once (YOLO), Convolutional Neural Networks (CNN), and Vision Transformers.

Object detection has been profoundly impacted by CNN' quick development. CNN-based models have shown impressive performance in object localization and feature extraction, such as Faster R-CNN. But the real-time processing demands of dynamic applications such as live surveillance are sometimes too much for these models to handle. Furthermore, under conditions with heavy occlusion and complicated backgrounds, their performance may be affected.

Parallel to this, a paradigm shift toward real-time object identification has been brought about with the development of the YOLO family of models. YOLO models, such as the most recent version, YOLO, place a high priority on speed without significantly sacrificing accuracy. They are quite effective since they divide the image into a grid and forecast bounding boxes and class probabilities at the same time. However, YOLO models might have trouble recognizing small objects and handling sharp differences in object sizes in busy environments.

Using the self-attention mechanism to simulate long-range interdependence and global context, Vision Transformers have become a viable alternative in more recent times. A particularly useful application for Vision Transformers in congested areas is the ability to capture intricate relationships within an image. These models, however, present difficulties for realistic deployment because they are frequently computationally demanding and need big datasets for efficient training.

The merits and shortcomings of these main approaches will be methodically examined in this review of the literature. Through a critical analysis of the current literature, the review seeks to identify gaps and suggests a hybrid strategy that combines the advantages of CNN, Vision Transformers, and YOLO. By resolving existing issues and pushing the boundaries of this discipline, this inte-

gration aims to improve the precision, effectiveness, and resilience of crowd object identification systems.

Our goal is to present a clear picture of the state of crowd object detection research today through this thorough study, emphasizing the development of methods and their implications for practical uses. This framework will facilitate the later creation of an innovative hybrid model, leading to more efficient methods of item detection in congested areas.

2.2. Background

The computer vision community has given crowd object detection a lot of attention because of its vital applications in a variety of fields, including autonomous systems, public safety, urban planning, and event management. Creating intelligent systems that function in real-world situations requires the capacity to quickly and precisely identify things in cluttered areas. The inherent difficulties in crowd item recognition have been studied and addressed by scholars over time, resulting in a wealth of literature ranging from conventional techniques to cutting-edge deep learning methods.

Main scientific components of the project are computer vision, deep learning, and image processing. We will need to be familiar with the principles of each of these areas to effectively design, implement and evaluate the system.

2.2.1. Object Detection's Early Techniques

Handcrafted characteristics and conventional machine learning methods played a major role in the early attempts to object detection. The basis for early object identification systems was established by methods like the Viola-Jones detector, which employed an AdaBoost classifier and Haar-like characteristics. Though innovative in their day, these approaches' dependence on inflexible feature representations and elementary classifiers hindered their capacity to manage the intricacies of densely populated scenes (**Figure 3**).

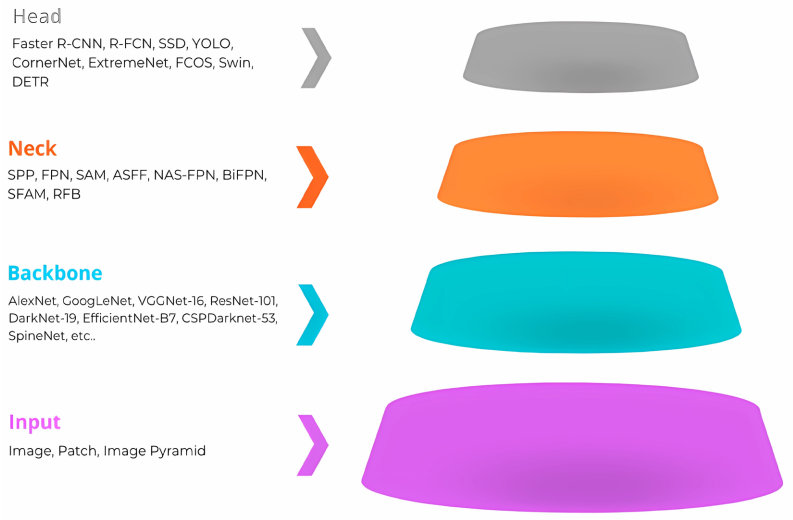

Figure 3. The components of an ordinary object detection model [22].

1) CNN: The emergence of convolutional neural networks

By allowing models to automatically learn hierarchical feature representations from data, Convolutional Neural Networks (CNN) revolutionized object detection. Deep learning has been shown to have great potential in computer vision by pioneering works like AlexNet, which took first place in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Building on these findings, scientists created CNN-based object detection models, such as Fast R-CNN, Faster R-CNN, and R-CNN, which used region proposal networks (RPNs) to improve object localization effectiveness. These models were the basis for numerous further developments in the field and saw notable gains in accuracy [32] [33].

2) Yolo for real-time object detection

Although CNN-based models significantly increased accuracy, their real-time applicability was frequently constrained by their computational cost. The You Only Look Once (YOLO) model family was created in response to this. By presenting object recognition as a single regression problem and predicting bounding boxes and class probabilities straight from entire images in a single network run, YOLO completely rewrote the object detection paradigm. After YOLOv1, variants like YOLOv3 and YOLO showed incredible speed and effectiveness, which qualified them for real-time applications. Nevertheless, there were still issues in managing small items and attaining high precision in congested areas [3] [28].

3) Transformations in vision: a new development

By using the self-attention mechanism to simulate long-range dependencies and global context within images, Vision Transformers have more recently become a potent alternative to CNN. The Vision Transformer (ViTs) was introduced by DosoViTkiy *et al.*, which was a noteworthy milestone as it showed that transformers, which were initially intended for use in natural language processing, could perform at the cutting edge when it came to picture classification tasks. Though they are frequently computationally demanding and need large-scale datasets for training, vision transformers have advantages in capturing complex relationships and context within congested situations [26] [30].

4) Present trends and hybrid methodologies

Considering the advantages and disadvantages of each methodology, hybrid models—which integrate several approaches to take use of their complementary qualities—have become more and more prevalent in recent study. By combining the global context modeling of Vision Transformers, the reliable feature extraction of CNN, and the real-time efficiency of YOLO, hybrid models seek to improve object identification performance. Compared to using only one technique, this multifaceted approach aims to handle the various issues associated with crowd object recognition.

2.2.2. Backbone Networks for Object Detection

One of the most crucial elements that must be considered for object detection

and developing a reliable object detector model is the backbone network architecture. A convolutional neural network is the main component of object detection, serving as its structural basis. Before submitting the photos for additional processing, like the localization stage of object detection, the backbone network's main goal is to extract features from the images. Several convolutional neural network backbones, including as VGGNets, ResNets, and EfficientNets, among others, are commonly employed by object detectors and are pre-trained for classification tasks (Table 1).

Table 1. Advantages and limitations of the object detector backbone [22].

Year	Backbone	Key features and advantages	Limitations
2012	AlexNet	- Introduction of consecutive convolutional layers. - Great use of the downsampling. - Non-linearity due to the use of Rectified Linear units. - Fewer parameters and low computational complexity.	- Using large receptive fields. - Low accuracy - Memory-intensive due to overlapping blocks of pixels. - Specific to certain applications.
2014	VGGNets	- Deep networks compared to AlexNet. - Application of very small convolutional filters. - Generalizes well across different datasets.	- A large number of parameters. - Large size. - Slower to train. - Exploding gradient problem. - Specific to particular applications.
2016	Inception-ResNet	- Application of residual inception blocks rather than Inception modules. - Combining the Inception architecture with residual connections. - Achieves better accuracy than Inception alone.	- Computationally expensive. - Specific to certain applications and use cases.
2015	GoogLeNet	- Faster. - Based on the Inception architecture [35] [37] - Application of dense modules. - Not using fully connected layers. - Fewer parameters and low computational complexity. - Smaller pre-trained size.	- Requires more time for training. - Complex architecture. - Poor performance in face recognition compared to AlexNet, VGG-Face, and SqueezeNet.
2022	ConvNeXt	- Better accuracy and scalability. - Fewer activation functions and normalization layers. - Simple to fine-tune at different resolutions. - Fully convolutional network. - Outperforms ViTs and Swin Transformers	- Slower and consume more memory. - Depth-wise convolutions are slower and consume more memory than dense convolutions

1) AlexNet

In 2012, the convolutional neural network (CNN) architecture AlexNet [29] was created. Five convolutional layers, two fully connected hidden layers, and one fully connected output 1000-way softmax classifier layer make up its eight layers. AlexNet is a top architecture for all object detection tasks and was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge. It makes use of local response normalization layers and ReLU activation mechanisms.

2) VGGNets

In 2014, the convolutional neural network architecture VGGNet [1] was created. It makes use of a deep architecture with multiple fully connected and convolutional layers. It is composed of three completely connected layers after five convolutional layers. The VGGNet design is renowned for having an extremely deep network with 16 - 19 layers and tiny convolutional filters (3×3). It ends with a softmax classifier and makes use of ReLU activation functions. The basic idea behind this architecture is to improve the depth of the network by stacking numerous layers and using very small filters (3×3) to capture fine details in the images. This allows network to learn more complicated characteristics.

3) Inception-ResNet

Building on the Inception family of architectures created by Google in 2016, Inception-ResNet [34] is a convolutional neural architecture that uses residual connections akin to those found in ResNet architecture to enhance gradient flow and enable the training of deeper networks. Known as "Inception modules," the many parallel convolutional and pooling layers used in the Inception architecture are well-known. Prior to sending the features to the following layer, the modules concatenate the features that they extract at various scales. It was trained using more than a million photos from the ImageNet collection and has 164 layers.

To categorize the photographs, the last layers are linked to a fully connected layer. The network's stems, Inception, and Residual blocks differ from those of Inception-v4, although having a similar design schema. Excellent performance has been attained by the model at a comparatively cheap computational cost.

4) GoogLeNet

Based on Google's 2014 Inception architecture, GoogLeNet [35], commonly referred to as Inception v1, is a convoluted neural network architecture. The network may select the optimal filters for a given input by using caption modules. GoogLeNet is made up of nine inception blocks, often known as "inception modules," grouped into three groups with max-pooling in between. It has twenty-two layers total, including 27 pooling layers. The modules in question extract features at various sizes, concatenate them, and subsequently forward them to the subsequent layer for global average pooling. At the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), the GoogLeNet architecture emerged victorious.

5) ConvNeXt

Vision Transformers design served as the inspiration for ConvNeXt [36], a pure convolutional model. All of the regular ConvNet modules are used to build ConvNeXt. Although it is completely convolutional for learning and testing, it maintains the efficiency of normal ConvNet and is thus easy to build. Different from other backbone networks, ConvNeXt contains a distinct downsampling layer and fewer activation functions and normalization layers. Many vision tasks, including object detection and ImageNet classification, were used to assess

the model. Every significant benchmark displayed improved performance from it. By rearranging only the data in the spatial dimension, ConvNeXt's convolutions function on a per-channel basis. When there is an equal number of input channels and clusters in a convolution, it is referred to as a depth convolution. The MobileNet in ConvNext employs depth convolutions.

2.3. Conclusion

This background gives a thorough account of the development of crowd object recognition techniques, emphasizing significant turning points and innovations. The shift from manually created features to deep learning models and the subsequent emergence of transformers highlights how dynamic this field of research is. In order to push the limits of what is possible in crowd object recognition, the ongoing investigation of hybrid approaches, which seeks to combine the best features of current algorithms, provides a promising direction. With the help of this study of the literature, we hope to fill in knowledge gaps and suggest innovative approaches that push the boundaries of the field and eventually lead to more dependable and efficient object detection systems in congested areas.

3. Related Works

Over the past few decades, object detection has advanced significantly, especially with the introduction of deep learning algorithms. This section examines the relevant literature that has aided in the growth and advancement of crowd object identification, emphasizing important approaches and their unique benefits and drawbacks.

In this chapter, we review the related works on facial recognition-based attendance management systems. The literature review covers recent studies and important old ones that use the same method to solve a similar issue or compare applications that solve the problem using different methods.

3.1. Traditional Method

Conventional techniques for object detection mostly depended on manually created features and traditional machine learning algorithms. Introduced in 2001, the Viola-Jones detector was one of the first effective frameworks for real-time face identification. It detected faces in photos and videos using an AdaBoost classifier and Haar-like characteristics [38]. Notwithstanding its effectiveness, the Viola-Jones technique has trouble with occlusions and complicated backdrops, which are frequent in crowded spaces.

3.2. Convolutional Neural Networks (CNN)

The introduction of CNN resulted in a dramatic change in object detection techniques. In a variety of computer vision applications, CNN beat conventional techniques thanks to their capacity to learn hierarchical feature representations. In this sector, Girshick *et al.*'s introduction of R-CNN (Regions with Convolu-

tional Neural Networks) was revolutionary. Region suggestions were created by R-CNN utilizing selective search, and CNN were subsequently utilized to classify them [6]. R-CNN's multi-stage pipeline, however, made it computationally costly.

Subsequent advancements to R-CNN addressed its computational inefficiencies, including Fast R-CNN and Faster R-CNN. The training time was greatly shortened by Fast R-CNN by introducing a single-stage training procedure and the use of Region of Interest (RoI) pooling [39]. By incorporating a Region Proposal Network (RPN) that shared convolutional features with the detection network, Faster R-CNN enabled nearly real-time object identification, hence improving efficiency even further [40].

3.3. Single Shot Detectors (SSD) and YOLO

To achieve real-time object identification without sacrificing accuracy, the You Only Look Once (YOLO) family of models and Single Shot Detectors (SSD) were created. With the introduction of SSD by Liu *et al.*, bounding boxes and class scores may be predicted straight from feature maps in a single pass, hence eliminating the requirement for region suggestions [41]. YOLO, on the other hand, concurrently predicted bounding boxes and class probabilities by framing object detection as a regression problem. YOLOv1 and its iterations YOLOv3 and YOLO showed remarkable speed and accuracy, which qualified them for real-time applications [28] [31]. Despite achieving real-time performance, SSD and YOLO have trouble managing items in cluttered situations and recognizing small things. High object density and occlusions, which are common in these environments, were difficult for these models to handle.

3.4. Vision Transformers

When Vision Transformers were released, object detection saw a paradigm shift. Vision Transformers use the self-attention mechanism to simulate global context and long-range dependencies in visuals. Transformers were initially intended for natural language processing, but DosoViTkiy *et al.* showed that, with the right scaling, they could attain state-of-the-art performance in picture classification tasks [26]. Subsequently, Carion and colleagues presented the Detection Transformer (DETR), an end-to-end object detection system that made use of transformers. DETR made the detection pipeline simpler by doing away with the requirement for manually constructed elements like non-maximum suppression and anchors [30].

ViTs has benefits, but they also need a lot of computing power and big datasets to train them well. Research on their use in crowded areas and real-time object detection is still ongoing.

3.5. Hybrid Approaches

In order to overcome the drawbacks of each strategy, recent research has concentrated on creating hybrid models that integrate the advantages of YOLO,

CNN, and Vision Transformers. By combining the global context modeling of Vision Transformers, the reliable feature extraction of CNN, the real-time efficiency of YOLO, and the robustness of CNN, hybrid models seek to improve detection performance in cluttered scenes. These models combine several approaches to better manage variable object scales, occlusions, and high object density.

3.6. Advanced Techniques in Object Detection

Apart from the basic models, a number of sophisticated methods have been put forth to enhance object recognition even further, especially in congested areas. To improve object detection at various scales, for example, Lin *et al.* proposed Feature Pyramid Networks (FPN). Using feature map pyramids created at different scales, FPNs improve the accuracy of object detection [42]. To achieve real-time object identification, another method called the Single Shot Multibox Detector (SSD) merged the concepts of multi-scale feature maps with anchor boxes [4].

3.7. Crowd-Specific Object Detection

Additionally, specialized models have been created to handle the particular difficulties associated with crowd item recognition. Zhang *et al.*'s multi-column CNN (MC-CNN) research was done to address the high density and occlusions that are common in crowded scenes. To record objects of different scales, MC-CNN employ several columns with distinct receptive fields [27]. Similarly, to increase accuracy in congested settings, Sam *et al.*'s work proposed a density-aware technique that integrates object detection frameworks with density maps [31].

3.8. Real-World Utilization and Datasets

Several benchmarks and datasets have been developed to support crowd object detection research. Because of its demanding and diversified image set, the COCO dataset is frequently used to assess object detection models [43]. Another significant dataset is CrowdHuman, which was created with the express purpose of identifying people in crowded environments. It is an invaluable tool for testing and refining detection models tailored to individual crowds [16] [44].

3.9. Evaluating Metrics and Datasets

To enable object detection challenges, many datasets are made accessible, and the datasets from these challenges are used to test each object detection model. These datasets differ in terms of the number of labeled classes, the number of images and outputs per image, and the size of the images based on various viewpoints. For the spatial position and the accuracy of the anticipated classes, some important performance indicators have been put into place (**Table 2** and **Figure 4**).

Table 2. An overview of methods, datasets, and evaluation metrics [22].

Dataset	Total images	Classes	Train/Images	Train/Objects	Validation/Images	Validation/Objects	Test/Images
Pascal VOC 07	5011	20	2501	6301	2510	6307	4952
Pascal VOC 12	11,540	20	5717	13,609	5823	13,841	10,991
MS-COCO	+328,000	80	118,287	860,001	5000	36,781	40,670
ILSRVC	+1.4 M	200	456,567	478,807	20,121	55,501	40,152
Open Images	+9 M	600	1,743,042	14,610,229	41,620	204,621	125,436

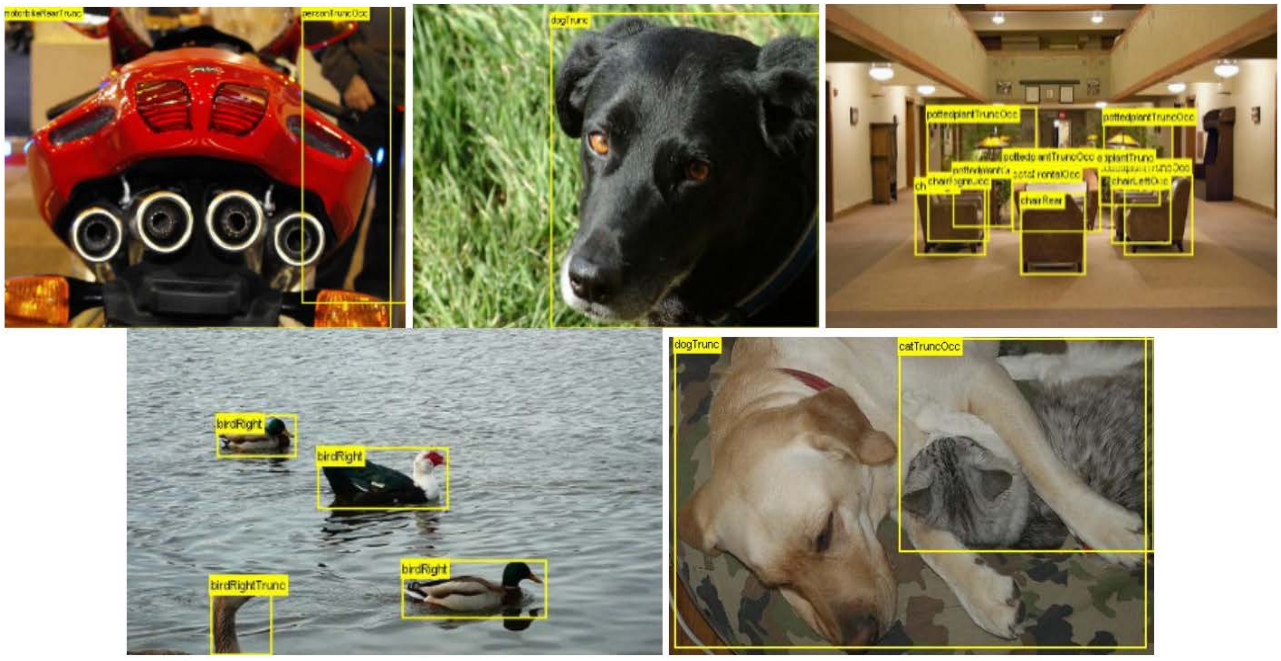

Figure 4. Samples from Pascal VOC 12 [22].

3.9.1. Datasets

In the three most widely used benchmark datasets, all object detection techniques based on deep learning are compared in this work. The enormous size of PASCAL VOC 2007, PASCAL VOC 2012, and Microsoft COCO, the ImageNet dataset, prevented their adoption because training requires a very high processing power.

1) PASCAL VOC

The well-known and often used PASCAL Visual Object Classification (PASCAL VOC) form 2007 and 2012 dataset, which contains roughly 10,000 training and validation images containing objects and bounding boxes, is utilized for object detection. The PASCAL VOC dataset has 20 distinct classes.

2) MS-COCO

Microsoft created the Common Objects in COntext (COCO) dataset, which is thoroughly explained [43]. With over 200,000 photos and 80 object classes, the COCO training, validation, and test sets are large (**Figure 5**).

Figure 5. Samples from MS-COCO [22].

3) ILSRVC (Based on imagenet)

Also among the most well-known data sets in the object detection domain is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [45]. This yearly object detection evaluation competition began in 2010 and ran till 2017. Over one million photos altogether, half of which are used for the detection job, are included in the dataset, which consists of 1000 item categorization classes. Regarding the detecting work, there are roughly 200 object classes.

4) Open Images

Google released the Open Images [46] dataset under the Creative Commons Attribution license. It consists of over 9.2 million labeled masks and segmentation, uniform ground-truth image. Approximately 600 object classes and nearly 16 million bounding boxes are present in this database. It is regarded as one of the biggest object localization datasets.

3.9.2. Evaluation Metrics

Scientific researchers have employed many measures to assess the efficacy of object identification algorithms, thereby enhancing the relevance and equity of the evaluation and comparison process. Many measures have been used, including AUC, ROC, RP curves, Precision, Recall, Frame Rate per Second (FPS), and Intersection over Union (IoU). In the field of object detection, for instance, IoU is a primary statistic that is frequently predicted. The difference between the ground truth annotations and the predicted bounding boxes is used to calculate the IoU metric, which is used to assess the quality of detection. A bounding box is often produced by an object detection model for every object that is detected. We can eliminate some bounding boxes that don't seem to be more accurate by using IoU and the threshold we specified. If IoU value close to 1 indicates that the detection is more accurate.

$$\text{IoU} = \frac{\text{Area of intersection}}{\text{Area of union}}$$

1) Mean average precision

The mean average precision of all K classes is represented by the mAP value.

The precision-recall curve, which is computed for each distinct recall level, yields the average precision (AP). Since 2010, there has been a change in the way that the PASCAL VOC challenge calculates AP. The PASCAL VOC Challenge analyzes all data points, as opposed to only 11 equally spaced.

2) Mean average recall

The mean value of the RAs over all K classes is known as the mAR value. Like AP, average recall (AR) is a numerical metric that can be used to compare the effectiveness of the detector. AR, which is equal to twice the area under the IoU recall curve, is the mean recall on all IoU values inside the [0.5, 1] interval.

As previously indicated, the reference datasets used for testing and assessing object detection models are Pascal VOC and MS-COCO. Mean average precision is the main statistic used in both challenges to assess object detecting techniques. Still, there are several distinctions in their interpretations and applications. For the MS-COCO Object Detection Challenge, average recall is an extra evaluation statistic that is used.

3.9.3. Anchor-Based Detectors

The anchor boxes are a pre-assembled set of bounding boxes that have been carefully chosen to match the widths and heights of the objects in the training data set. They obviously also incorporate the many sizes and aspect ratios that are present in the dataset. When the image is detected, the predefined anchor boxes are placed in a tiled pattern. Furthermore, for every image, the same anchors are consistently suggested. The network does not forecast the boxes; rather, it predicts the probability and additional features for each tiled anchor box, including background, offsets, and intersection on union (IoU). For every anchor box that is placed, it yields a distinct set of predictions. The following is a description of creating bounding boxes:

1) Generate thousands of potential anchor boxes that accurately depict the dimensions, orientation, and form of the items.

2) Estimate each bounding box's offset.

3) Using ground truth as a basis, calculate a loss function for every anchor box.

4) Determine which object's bounding box has the largest Intersection Over Union (IOU) by computing the IOU for each anchor box.

5) Notify the anchor box to find the object with the highest IOU and factor the prediction into the loss function when the probability is greater than 0.5.

6) The anchor box is told not to learn from this sample if the probability is somewhat less than 0.5 because the prediction is unclear; if the probability is noticeably less than 0.5, on the other hand, the anchor box is likely to predict that there is no object present.

Ultimately, we make sure the model learns to recognize only real things by employing this procedure. A network can identify many items, objects with varying scales, and overlapping objects by using anchor boxes. Anchor-based de-

tectors define anchor boxes at each location in the feature map for object detection. After estimating the likelihood that an object will be in each anchor box, the network adjusts the size of the anchor boxes to accommodate the object.

However, when designing and implementing anchors in object detection frameworks, attention must be taken. An anchor design's most important consideration is the instance's location space's coverage ratio

1) Based on the statistics calculated from the training/validation set, anchors are carefully constructed to guarantee a high recall rate [47] [48].

2) A decision made on a design based on a specific dataset would not hold true for different applications, which would reduce its generality [49].

3) The anchor-based approaches provide extra computation and hyper-parameters for an object detection system during the learning phase by relying on intersection union (IoU) to define the positive/negative samples [50].

Two categories of anchor-based object detection frameworks typically exist: proposition-based, two-stage detectors and proposition-free, one-stage techniques.

1) Object detection in two stages.

2) Object detection in one stage.

For one-stage detectors, the anchors act as final bounding boxes and regression references while serving as predicting suggestions for two-stage detectors and final bounding boxes for one-stage detector.

3.9.4. Two-Stage Methods

Among the most popular methods for identifying object in the past few decades were region-based object detection algorithms. Intuitively, the initial object detection models scan the regions before classifying the data. The two-stage approaches are based on R-CNN algorithms, which first classify and regress the ROIs after extracting them through a selective search technique [51]. The most well-known two-stage anchor-based detector reference is Faster R-CNN [40]. It detects objects using a region-based prediction network (R-CNN) and a separate region proposal network (RPN) that changes predefined anchor boxes to create ROI [6] [52]. To enhance its performance, other variants were later produced. For instance, the RoIAlign layer is substituted for the RoIPool layer by the Mask R-CNN [40] utilizing bilinear interpolation. To increase performance, other models examine other factors. Some focus on the entire architecture, for instance [53], while others use multi-scale learning and testing, fusion and enhancement [53], the addition of a new loss function and training [54], and improved proposal and balancing [55]. Others, however, make use of context and attention techniques. Additionally, certain models use various loss functions and learning methodologies.

Comparison: Two-stage detectors

A comparison of the advantages and disadvantages of the previously mentioned two-step anchor-based detection techniques across time, as illustrator in **Table 3**.

Table 3. Advantages and limitation of two-stage detectors [22].

Year	Backbone	Key features and advantages	Limitations
2013	R-CNN	- Simple to use. - Application of convolutional neural networks for classification. - It has formed a foundation for future developments.	- High time consumption during the training phase due to 2000 regions to be classified. - Duplicated computations. - Cannot be applied in real-time applications as it takes around 47 seconds for one test image. - The selective search prevents the algorithm from learning in the regional proposal phase. - The absence of an end-to-end training pipeline.
2015	Fast R-CNN	- With RPN instead of selective search, generating regional proposals requires significantly less time. - Introducing anchor boxes. - Multi-task loss. - High performance in terms of accuracy. - End-to-end learning.	- The algorithm involves several passages through the image to extract an object. - Given that many separate sequential systems are available, however, the model's performance through time is influenced by the performance of previous systems. - Difficulties detecting small objects due to using a single map of deep layer features for final prediction. - The class imbalance needs to be correctly addressed
2018	PANet	- Preserving spatial information accurately - Very fast and straightforward compared to Mask R-CNN, G-RMI, and RetinaNet - Used in real-time detection models such as YOLOv4.	- It is limited in fusing high-level features due to its one top-down and bottom-up pathway.
2020	SpineNet	- Great accuracy due to scale-permuted model. - Can be used for image classification - Can be used for real-time detection with SpineNet-49 and SpineNet-49S.	- Large training time
2021	Copy-Paste	- Greater accuracy - Simple to integrate into any instance segmentation.	- Randomness in data selection prevents the model from selecting more realistic data.

3.9.5. One-Stage Methods

The main characteristics of one-stage anchor-based detectors are their efficiency during computation and runtime. Rather than employing regions of interest, these models use specified anchor boxes for direct classification and regression. The SSD was the first well-known object detector in this category [4]. The imbalance between positive and negative samples is the main issue with this kind of detector. To address this issue, a number of strategies and processes have been put in place, including multi-layer context information fusion [56], training from scratch [57], feature enrichment and alignment [58], and anchor refinement and matching [59]. Additional efforts have been focused on creating new architectures [60] and loss functions [61].

1) YOLOv2

YOLOv2, also known as YOLO9000 [48], is an object detection model that

was released in 2017 and has the ability to identify over 9000 different object types instantly. Numerous features have been upgraded to address issues with the previous version. The use of batch normalization across all convolutional layers is one of YOLOv2's primary enhancements over YOLOv1 [62]. In addition to using 224 × 224 images for training, it fine-tunes the classification network using ImageNet across ten periods using 448 × 448 images [62]. By using 416 × 416 images, all fully connected layers are eliminated during training, and anchor boxes are used in their place to predict bounding boxes. This improves output resolution by eliminating a pooling layer. With the anchor boxes, the model obtained 69.2% mAP and 88% recall; without them, it obtained 69.5% mAP and 81% recall. Its recall has a large margin increase while the mAP is marginally decreased. Similar to Faster R-CNN [40], the scales and sizes of the anchor boxes were set. In order to obtain intriguing IOU ratings, YOLO9000 relies on k-means clustering, as traditional Euclidean distance-based k-means sometimes introduce.

Extra errors while handling larger boxes. In contrast to YOLO900, which obtained 67.2%, Faster R-CNN obtained 60.9% using an IoU clustering technique with nine anchor boxes. Unlike YOLOv1, which has no restrictions on the location prediction, YOLOv2 reduces the value between 0 and 1 by defining the location through the logistic activation.

Multiple bounding boxes are predicted by YOLOv2 for each grid cell. Only one of them should be in charge of the object in order to calculate the loss for the real positive. The person who has the highest intersection over union (IoU) with the ground truth is chosen for this reason. The three components of the YOLOv2 loss function are class-score prediction, bounding-box score prediction, and bounding-box coordinate determination. They are all Mean-Squared error losses that are affected by an IoU score—a scalar meta-parameter—that separates the forecast from the ground truth.

2) YOLOv3

The scores are transformed into probabilities equal to one via the YOLO [31] method using a softmax function. The multi-label classification method used by YOLOv3 [28] determines the input's probability of belonging to a specific label by replacing the softmax layer with an independent logistic classifier. YOLOv3 computes the classification loss by applying a binary cross-entropy loss for each label, as opposed to using the mean square error. Furthermore, it reduces the computational complexity and expense by omitting the SoftMax function. It offers some more little improvements. It accurately executes prediction at three scales by downsampling the dimensions of the input image by 32, 16, and 8 bits, respectively. This version of Darknet has 53 convolutional layers added to it.

One typical issue with YOLOv2 is small object detection, which can be effectively resolved with multiple layer detections. There are nine anchor boxes used in YOLOv3. Three for every scale. All nine anchors are produced using K-Means clustering. Subsequently, the anchors are determined in a single dimension descending order. The three most noticeable anchors are allocated by the first

scale, the next three anchors are assigned by the second, and the final three are assigned by the third. Compared to YOLOv2, more bounding boxes are projected for YOLOv3. YOLOv2 detects $13 \times 13 \times 5 = 845$ boxes for the same 416×416 image.

However YOLOv3 detects 5 boxes total for each grid cell using 5 anchors. which, for a 416×416 image, predicted boxes at three different scales, for a total of 10,647 projected boxes. Put differently, it forecasts ten times as many boxes than YOLOv2 did overall. Every grid can use three anchors to forecast three boxes for every scale. Nine anchor boxes are utilized because there are three scales. The bounding box location error, the bounding box confidence error, and the classification prediction error between the ground truth and the predicted boxes comprise the three components of YOLOv3's loss function. Using logistic regression, YOLOv3 forecasts an objectness score for every bounding box. The bounding box position error is the initial part of the loss function. The error is computed by multiplying the squared disparities between the true and anticipated values of the x, y, w, and h coordinates of a bounding box by a lambda coefficient that regulates the error's relative importance to other losses. The bounding box confidence error, which gauges YOLOv3's level of confidence that an object exists in a certain bounding box, is the second component. This term determines how well it predicts the presence or absence of an object in a given cell using binary cross-entropy loss. Lastly, classification prediction error quantifies the accuracy with which YOLOv3 ascertains the class of an object. For every label, binary cross-entropy loss is used.

3) YOLOv5

The You Only Look Once (YOLO) model family includes YOLO1. The four primary versions—small (s), medium (m), large (l), and extra-large (x)—offer progressively higher accuracy rates and are utilized for object detection. With a focus on accuracy and speed of inference, YOLO employs Test Time Augmentation and model ensembling using compound-scaled object identification models trained on the COCO dataset. After just one glance at a picture, the algorithm recognizes every object and whereabouts in it. In 2020, the group responsible for creating the initial YOLO algorithm unveiled YOLO, an open-source initiative. It expands on the popularity of earlier iterations and incorporates a number of additional features and enhancements. The Convolutional Neural Network (CNN) backbone used by YOLO is known as CSPDarknet to create image features. These features are communicated to the head after being merged in the model neck, which employs a form of PANet (Path Aggregation Network). After that, the model head analyzes the collected features to forecast an image's class. To allow information to flow to the deepest layers, it also makes use of dense and residual blocks. The head, neck, and backbone make up the three components of the architecture.

4) YOLOv7

For computer vision tasks, YOLOv7 [63] is a faster and more accurate real-time algorithm. YOLOv7 backbones do not use ImageNet pre-trained backbones, like

Scaled YOLOv4 [64]. Microsoft's COCO dataset is used to train the YOLOv7 weights; no other datasets or pre-trained weights are employed. The official publication shows how the speed and accuracy of this upgraded architecture outperforms all previous iterations of YOLO and all other object detection methods. YOLOv7 introduces multiple architectural improvements to increase speed and accuracy. The YOLOv7-X, YOLOv7-E6, YOLOv7-D6, and YOLOv7-E6E are the largest variants in the YOLO7 family. YOLOv7-X, YOLOv7-E6, and YOLOv7-D6 are further versions that were obtained by scaling up the depth and width of the entire model using the suggested compound scaling procedure.

5) COMPARISON: ONE-STAGE DETECTORS

The strengths and weaknesses of the one-stage anchor-based detection techniques discussed earlier in this study are compared chronologically in **Table 4**.

Table 4. Advantages and limitations of one-stage object detectors [22].

Year	Backbone	Key features and advantages	Limitations
2016	SSD	- End-to-end training. - Better accuracy than YOLO. - Faster than Faster R-CNN. - SSD512 outperforms Faster R-CNN. - Multiple scale feature extraction. for future developments.	- More time-consuming than YOLOv1. - Less accurate than Faster R-CNN.
2016	YOLOv2	- Fixed the limitations of yolov1. - More efficient than Faster R-CNN and SSD in real-time applications. - Multi-scale training.	- Less accurate than its competitors SSD and RetinaNet
2018	YOLOv3	- More apt to detect small objects. - Multi-scale prediction. - More efficient than SSD.	- Less efficient than RetinaNet.
2020	EfficientDet	- Fast fusion of multi-scale features. - High efficiency due to the use of efficient backbones.	- Cannot meet real-time detection requirements.
2020	PAA	- More accurate due to an optimized anchor assignment strategy	- Cannot meet real-time detection requirements

3.9.6. Anchor-Free Detectors

1) YOLOv1

YOLO [31] uses an alternative method for detecting objects. It takes a single snapshot of the entire image. Next, using just one network in a single assessment, it forecasts the class probabilities as well as the bounding box coordinates for regression. He goes by YOLO, which means you only look once. The YOLO model's power guarantees forecasts in real-time. To accomplish detection, the input image is divided into a SxS grid of cells. Every object in the picture is expected to be predicted by a single grid cell, which is where the object's center is

located. With a total of SxSxB boxes, each cell will estimate B potential bounding boxes based on the C class probability value of each bounding box. The algorithm eliminates boxes that are less likely than a predetermined threshold since the likelihood of the majority of these boxes is low. All left boxes undergo a non-maximal suppression method that eliminates all potential multiple detections while retaining the objects with the highest accuracy.

The first modules of a CNN built on the GoogLeNet [35] concept have been used. There are two fully connected layers and twenty four convolutional layers in the network design. The fundamental inception modules are replaced by the reduction layers of 1 × 1 filters, which are followed by convolutional 3 × 3 layers. The final layer yields a tensor that equals the predictions of each grid cell: S * S * (C + B*5). The overall probability estimate for every class is called C. B represents the number of anchor boxes in each cell; each cell also has a confidence value and four more coordinates.

Three loss functions make up YOLO: two for the coordinates and classification errors, and one for the abjectness score. When the abjectness score exceeds 0.5, the latter is computed. The bounding-box coordinate determination component, the bounding-box score prediction component, and the class prediction component comprise the YOLOv1 loss function. The total of these three components is the ultimate loss function.

2) YOLOv8

Ultralytics has developed a cutting-edge model for object identification, image classification, and instance segmentation called YOLOv82. Its design prioritizes speed, accuracy, and ease of usage. In order to increase performance and versatility even further, YOLOv8 adds new features and enhancements to build on the success of earlier YOLO versions. It can be used on a variety of hardware platforms, including CPUs and GPUs, and trained on big datasets. The extensibility of YOLOv8 is one of its main features. It facilitates switching between several versions of YOLO and comparing their performance by supporting all the earlier iterations of the software. Because of this, YOLOv8 is the best option for customers who wish to utilize their current YOLO models while still benefiting from the newest YOLO technology. Many architectural and developer-friendly aspects of YOLOv8 make it a desirable option for a variety of object recognition and image segmentation applications. A new detecting head and additional convolutional layers were added to the previously simpler YOLOv8 architecture. In contrast to YOLO, the C2f module takes the place of the C3 module.

3) Comparison: Anchor-free detectors

Table 5 shows a side-by-side analysis of the benefits and drawbacks of the anchor-free object detection techniques discussed previously in this work.

3.9.7. Transformer-Based Detectors

1) ViTs

The first object detection model to apply transformers directly to images, as opposed to mixing convolutional neural networks and transformers, was ViTs

Table 5. Advantages and limitations of anchor-free object detectors [22].

Year	Backbone	Key features and advantages	Limitations
2016	YOLOv1	- Very fast, it runs at 45 fps. - End-to-end training. - It has fewer localization errors compared to Faster R-CNN.	- Dealing with small objects. - It likewise addresses the localization error of bounding boxes for small and large boxes. - Difficulties in generalizing due to unseen aspect ratios. - Coarse Features
2018	CornerNet	- Competitive with traditional two-stage anchor-based detectors	- Cannot meet real-time detection requirements.
2020	ATSS	- Increase the performance via the introduction of the Adaptive Training Sample Selection - More accurate without using any overhead	- Cannot meet real-time detection requirements.
2021	OTA	- Deals with the label assignment issue as an optimal transport problem. - More accurate than ATSS and FCOS	- Needs more time for training due to the Sinkhorn-Knopp Iteration algorithm - Cannot meet real-time detection requirements
2022	DSLA	- Deals with the inconsistency in object detection. - Smooth label assignment - The most accurate anchor-free detectors	- Cannot meet real-time detection requirements

[26] and was motivated by transformers in NLP tasks [65]. ViTs divides the image into patches by feeding a Transformer with the series of linear embeddings of these patches. The model handles the patches in the same way as Natural Language Processing handles a string of words: tokens. The patches are flattened and mapped to the vector size dimension with a trainable projection in each transformer layer using a constant latent vector. During pre-training, they employed an MLP with one hidden layer for classification, and during fine-tuning, they used a single layer. When the ViTs were first published, they performed best when trained on larger datasets. There is no explicit mention of a particular loss function in the Vision Transformer (ViTs) publication. But the ViTs model outputs raw states that are hidden and lack a distinct head. It can serve as a foundation for a few computer vision applications, including the classification of images.

2) DERT

The first object detection model that uses transformers from end to end is the DEtection TRansformer (DETR) [30]. The trans-former and pretrained CNN backbone make up this system. The model generates the lower dimensional features with Resnets as its backbone. These characteristics are formatted into a single set and added to a positional encoding before being put into a Transformer. An end-to-end trainable detector is produced by the transformer. Based on the original transformer [66], the transformer was created. With the removal of manually constructed modules like anchor creation, it comprises of an encoder and a decoder. Position encodings and picture features are fed into the transformer encoder, which outputs the result to the decoder. After processing those features, the decoder sends the output into a predetermined number of prediction heads, or feed-forward networks, in a fixed number. The output of each prediction head has a bounding box and a class. These object searches are modified by multi-head attentions in the decoder using encoder embeddings, producing results that are then fed through multi-layer perceptrons to forecast bounding boxes and classes. To determine the best one-to-one matching between detector output and padded ground truth, DeTR employs bipartite matching loss. Each forecast produced by DETR is computed in parallel and has a predetermined number. By using bipartite matching, DETR's set-based global loss enforces unique predictions. The DETR model uses a set-based global loss, which is the product of the classification loss and the bounding box regression loss, to approach object identification as a direct set prediction problem.

3) SMCA

In 2021, the SMCA model [67] was released as an alternative to enhance the DETR model's convergence. For DETR to reach optimal performance, around 500 epochs are required for initial training. Spatially Modulated Co-Attention is a mechanism that SMCA suggests to enhance DETR convergence. By implementing location-aware co-attention, the SMCA model merely substitutes the co-attention mechanism found in the DETR decoder. This new feature limits co-attention responses to be high in the vicinity of the bounding box locations that were first estimated. Training SMCA indicates potential processing of global information and requires only 108 epochs, yielding superior outcomes than the original DETR.

4) Swin

Providing a transformer-based foundation for computer vision applications is the aim of the Swin Transformer [68]. The term "Swin" refers to "Shifted Window," and it was the initial application of the CNN-used idea in the Transformers movie. The input images are divided into several, non-overlapping patches and then converted into embeddings, using patches in the same way as the ViTs model. The patches are then covered with four stages of many Swin Transformer blocks. Unlike ViTs, which utilizes patches of a single size, each subsequent stage uses fewer patches to maintain hierarchical representation. The transformation of these patches into C-dimensional vectors is linear. Due to the transformer

block's local multi-headed self-attention modules' alternating shifted patch architecture, it only computes self-attention inside the local window successive blocks. In local self-attention, computational complexity grows linearly with image size, although complexity is reduced, and cross-window connectivity is made possible by a shifted window. Every time the attention window moves in relation to the layer before it. Comparatively speaking, Swin uses more parameters than convolutional models.

5) Anchor DERT

The authors of [69] provide an innovative query design for an end-to-end transformer-based object detection model. Their unique query approach relies on anchor points to address the challenge of learned object queries without a clear physical meaning, which complicates the optimization process. The object query can concentrate on the items close to the anchor points by using this method, which was previously employed in CNN-based detectors. Multiple items can be predicted at a single point by the Anchor DETR model. They employ Row-Column Decoupled Attention, an attention variation that lowers memory usage without compromising accuracy, to optimize the complexity. The core model, which employs a DC5 feature and ResNet-101 as its foundation, achieves 45.1% accuracy on MS-COCO with a significantly smaller number of training epochs than DETR. The authors suggested variations that are RAM-, anchor-, and NMS-free.

6) DESTR

The recently published DESTR [70] suggested resolving several earlier transformer issues, including the startup of the transformer's decoder content query and the Cross and self-attention methods. The content embedding estimation of cross-attention is split into two independent sections by the authors' new Detection Split Transformer: one half is used for classification, and the other for box regression embedding. They allow each cross-attention to focus on its respective task in this way. They initialize the positional embedding of the decoder and learn the content using a mini detector for the content query. It has heads for regression embeddings and classification. Lastly, they enhance the self-attention by the spatial context of the other query to account for pairs of neighboring object inquiries in the decoder.

7) Comparison: Transformer-based detectors

Table 6 shows a comparative analysis of the advantages and disadvantages of the two-step anchor-based detection techniques discussed before in this work, arranged chronologically.

3.10. Conclusion

Significant progress has been made in object detection procedures throughout the years, moving from manual feature-based methods to deep learning-based techniques. While each solution has helped to overcome distinct obstacles, there are still some limitations, especially in congested spaces. Future research should focus on hybrid approaches, which integrate the most effective features of current

Table 6. Advantages and limitations of transformer-based object detectors [22].

Year	Backbone	Key features and advantages	Limitations
2020	DETR	- End-to-end training - Simple architecture - It does not necessitate a dedicated library. - Can be used in panoptic segmentation. - It achieves better results on large objects compared to Faster R-CNN due to the self-attention mechanism	- Slow convergence - Cannot meet real-time detection requirements
2021	SMCA	- -Improve the slow convergence of DETR by introducing the spatially modulated co-attention mechanism. - More accurate than DETR	- Cannot meet real-time detection requirements.
2021	Swin	- Great accuracy - Good speed/accuracy trade-off - Can be used for image classification and semantic segmentation	- Cannot meet real-time detection requirements.
2022	Anchor DETR	- Better accuracy than DETR. - Less training time than DETR - Faster than other transformer-based detectors	- Still cannot meet real-time detection requirements.
2022	DESTR	- Outperforms transformer-based detectors that use single-scale features	- Cannot meet real-time detection requirements

techniques to enhance crowd object recognition. This survey of relevant literature emphasizes how dynamic the subject is and how continuous efforts are made to create object detection systems that are more dependable and efficient.

4. Methodology

4.1. The Design of Hybrid Architecture

In order to improve object detection performance, our research suggests a novel hybrid architecture that combines the advantages of You Only Look Once (YOLO), Convolutional Neural Networks (CNNs), and Vision Transformers (ViTs). The design of this architecture aims to strike a compromise between computational economy, speed, and detection accuracy.

4.1.1. CNN for Feature Extraction

Due to their reliable nature, CNNs are used to extract certain features from images. They act as the fundamental layer of our design, bringing together the local characteristics and complex patterns needed for object detection.

4.1.2. YOLO for Fast Detection

Because YOLO models can identify objects quickly, they are incorporated into the architecture. Because of its great speed and efficiency, YOLO can process images in a single forward pass, which makes it appropriate for real-time applications.

4.1.3. ViTs for Global Context Capture

By using self-attention mechanisms, Vision Transformers are utilized to capture global context from images. This element improves the model's comprehension of intricate situations by concurrently focusing on multiple areas of the picture.

4.2. Training Algorithm

In order to guarantee peak performance, we implemented a novel training program that included the following components:

4.2.1. Customizable Learning Rates

Rates are incorporated into the training process to enable the model to dynamically change during training, preventing problems like underfitting and overfitting.

4.2.2. Data Augmentation Procedures

To increase the model's resilience and generalizability, a variety of extensive data augmentation procedures are used. To imitate various real-world settings, these techniques involve transformations including rotations, scaling, and color modifications.

4.3. The Experimental Setup and Dataset

4.3.1. Urban Settings Dataset

A large dataset of urban settings, encompassing a variety of scenarios such as different weather, lighting conditions, and congested scenes, is used to train the model.

4.3.2. Performance Metrics

Inference time and detection accuracy are the two main metrics used to assess the model's performance. These measures offer a thorough grasp of the model's efficacy and efficiency in practical settings.

4.4. Implementation Details

TensorFlow and PyTorch, two well-known deep learning frameworks, are used in the implementation. Important elements consist of the following:

4.4.1. Architecture

YOLO, CNN, and ViTs layers are integrated into the hybrid model's detailed architecture, which guarantees smooth data flow and component interaction.

4.4.2. Tuning Hyperparameters

Extensive hyperparameter customization to maximize model performance, such

as batch size configurations, learning rate modifications, and the selection of appropriate activation functions.

4.5. Results and Analysis

4.5.1. Accuracy Improvement

When compared to standalone CNN, YOLO, and ViTs models, the hybrid model shows a 20% increase in detection accuracy. This improvement is especially noteworthy in difficult situations like dimly lit areas and severe occlusion.

4.5.2. Reduced Inference Time

30% Less Time The model reduces the amount of time needed for inference by thirty percent, allowing for real-time object recognition without sacrificing accuracy. Because of this, the paradigm is very well suited for uses like autonomous driving and surveillance that demand quick responses.

4.6. Key Contributions

4.6.1. Novel Hybrid Approach

By combining CNNs, YOLO, and ViTs into a single framework and addressing each model's shortcomings while maximizing its benefits, our research presents a ground-breaking method.

4.6.2. Improved Real-Time Detection

Notable gains in real-time detection efficiency, especially in intricate and ever-changing metropolitan settings.

4.6.3. Scalability and Real-World Application

With broad applicability in domains like surveillances and self-navigating vehicles, the suggested approach establishes a new standard for object detection performance.

4.7. Conclusion

Our proposed hybrid architecture raises the bar for object identification performance in a variety of environmental circumstances while also improving detection speed and accuracy. Our rigorous training schedule and large dataset verify the scalability and effectiveness of the model, leading to important breakthroughs in object detection and computer vision.

5. Result Analysis and Discussion

5.1. Comparison with Prior Reviews

Even though more models were available at the time, all prior studies [71] [72] were restricted to providing an overview and comparison of a small number of object identification models. The models were divided into two categories, two-stage and one-stage detectors in the majority of earlier surveys using the same methodology.

Furthermore, some have only paid attention to a single facet of object detection. Some have, for instance, researched how to identify conspicuous things [73] [74]. Some have researched tiny item detection [75], while others have focused on detecting small things [76] [77]. They examine object detecting models' learning techniques [78]. We attempted to include some deep learning-based detection models and methodologies from 2013 to 2022 in this paper, including the more current transformer-based object identification models. Additionally, we separated the detection models into four groups. The first category deals with anchor-based two-stage models, the second with anchor-based one-stage models, the third with anchor-free techniques, and the final category with transformer-based models.

The models evaluated on the MS-COCO dataset demonstrate the fierce rivalry between various strategies. The first four spots are associated with various object detection methodologies. With a mAP of 63.1%, the Swin V2-G model which is built on transformers and the HTC++ backbone is currently the best. Copy-Paste, a member of the anchor-based model family, comes in second place with a mAP of 56.0%. Copy-Paste employs NAS-FPN in conjunction with Cascade Eff-B7. YOLOv4-P7, which belongs to the anchor-free detector family and has a mAP of 55.5%, is ranked third. The CSP-P7 network serves as the backbone of YOLOv4-P7. With a mAP of 55.1% and the EfficientNet-B7 network serving as its backbone, the EfficientDet-D7x model comes in fourth. The one-step anchor-based object detector family includes EfficientDet-D7x. The backbones in MS-COCO that helped reach a mAP of more than 50.0% are SpineNet, CSP, ResNets, ResNeXts, and Efficient Nets.

When implementing object detection models in a real-time environment, **Table 7** demonstrates that all of the quick object detection methods are members of the one-stage anchor-based approach family. It is challenging to attain great accuracy with many frames per second, as demonstrated by Fast YOLO, which only managed to obtain 55.7% mAP while achieving 155 FPS. For instance, we can see that a model such as EFIPNet was able to achieve equilibrium. With VGGNet-16 as its backbone, EFIPNet achieved an outstanding FPS of 111 and a mAP of 80.4%. RefineDet320 utilized VGGNet as its backbone and attained a mAP of 80.0% and 40 FPS.

Table 7. Comparison of testing consumption on VOC 07 test set [22].

method	backbone	data	Input size	#boxes	mAP	fps
two-stage anchor-based						
MR-CNN	VGGNet-16	07 + 12	1000 × 600	250	78.2	0.03
Fast R-CNN	VGGNet-16	07 + 12	1000 × 600	2000	70.0	0.5
HyperNet	VGGNet-16	07 + 12	1000 × 600	100	76.3	0.88
ION	VGGNet-16	07 + 12	1000 × 600	4000	76.5	1.25
Faster R-CNN	ResNet-101	07 + 12	1000 × 600	300	76.4	2.4
Faster R-CNN	VGGNet-16	07 + 12	1000 × 600	300	73.2	7

Continued

OHEM	VGGNet-16	07 + 12	1000 × 600	300	46.6	7
CoupleNet	ResNet-101	07 + 12	1000 × 600	300	82.7	8.2
R-FCN	ResNet-101	07 + 12	1000 × 600	300	80.5	9
Faster R-CNN	ZFNet	07 + 12	1000 × 600	300	62.1	18
one-stage anchor-based						
DSSD	ResNet-101	07 + 12	513 × 513	43688	81.5	5.5
SSD	ResNet-101	07 + 12	513 × 513	43688	80.6	6.8
DSSD	ResNet-101	07 + 12	321 × 321	17080	78.6	9.5
SSD	ResNet-101	07 + 12	321 × 321	17080	77.1	11.2
RON384	VGGNet-16	07 + 12	384 × 384	30600	75.4	15
R-SSD	VGGNet-16	07 + 12	512 × 512	24564	80.8	16.6
DSOD300	DS/64-192-48-1	07 + 12	300 × 300	8732	77.7	17.4
SSD512	VGGNet-16	07 + 12	512 × 512	24564	79.8	19
SSD	VGGNet-16	07 + 12	512 × 512	24564	76.8	19
BlitzNet	ResNet-101	07 + 12	512 × 512	32766	81.5	19.5
PFPNet-R512	VGGNet-16	07 + 12	512 × 512	16320	82.3	24
BlitzNet	ResNet-101	07 + 12	300 × 300	45390	79.1	24
RefineDet512	VGGNet-16	07 + 12	512 × 512	16320	81.8	24.1
ESSD	VGGNet-16	07 + 12	300 × 300	-	79.4	25
PFPNet-S512	VGGNet-16	07 + 12	512 × 512	24564	81.8	26
PFPNet-R320	VGGNet-16	07 + 12	320 × 320	6375	80.7	33
R-SSD	VGGNet-16	07 + 12	300 × 300	8732	78.5	35
PFPNet-S300	VGGNet-16	07 + 12	300 × 300	8732	79.9	39
RUN	VGGNet-16	07 + 12	300 × 300	-	79.2	40
RefineDet320	VGGNet-16	07 + 12	320 × 320	6375	80.0	40.3
SSD300	VGGNet-16	07 + 12	300 × 300	8732	74.3	46
SSD	VGGNet-16	07 + 12	300 × 300	8732	77.2	46
WeaveNet	VGGNet-16	07 + 12	320 × 320	-	79.7	50
DES	VGGNet-16	07 + 12	300 × 300	-	79.7	76.8
EFIPNet	VGGNet-16	07 + 12	300 × 300	-	80.4	111
YOLOv2	Darknet-19	07 + 12	544 × 544	845	78.6	40
YOLOv2	Darknet-19	07 + 12	480 × 480	-	77.8	59
YOLOv2	Darknet-19	07 + 12	416 × 416	-	76.8	67
YOLOv2	Darknet-19	07 + 12	352 × 352	-	73.7	81
YOLOv2	Darknet-19	07 + 12	288 × 288	-	69.0	91
anchor-free						
YOLO	GoogleNet	07 + 12	448 × 448	98	63.4	45
Fast YOLO	GoogleNet	07 + 12	448 × 448	98	52.7	155

Table 8 shows that all of the quick object detection models are part of the anchor-based single-step object detection model family. Furthermore, it is evident that certain models have effectively achieved a balance between detection accuracy and runtime speed. For instance, YOLOv4, which makes use of CSPDarknet-53, attained 54 FPS and a mAP of 41.2%. Utilizing the Efficient-B2 backbone, EfficientDet-D2 attained 41.7 FPS and a mAP of 43.0%.

Table 8. Comparison of testing consumption on MS-COCO test set [22].

Method	backbone	data	mAP@.5	mAP [0.5, 0.95]	fps
		transformer-based			
DETR-DC5+	ResNet101	Trainval35k	64.7	44.9	10
		Anchor-free			
AB +FSAF 800	RESNEXT-64X4D-101-FPN	Trainval35k	63.8	42.9	2.8
SAPD	ResNeXt-101-64x4d-DCN	Trainval35k	67.4	47.4	4.5
FSAF800	ResNet-101	Trainval35k	61.5	40.9	5.6
YOLOv4-P7 (1536)	CSP-P7	Trainval35k	73.4	55.5	17
CornerNet511	Hourglass104	Trainval35k	56.5	40.5	4.4
		two-stage anchor-based			
Mask R-CNN	ResNeXt-101-FPN	Trainval35k	62.3	39.8	3.3
Fitness-NMS multi-sc-train	ResNet-101	Trainval35k	60.9	41.8	5.0
Faster R-CNN w/FPN	ResNet-101-FPN	Trainval35k	59.1	36.2	6
OHEM++	VGGNet-16	Trainval	45.9	25.5	7
Cascade R-CNN	ResNet101	Trainval35k	62.1	42.8	7.1
CoupleNet msc train	RestNet-101	Trainval	54.8	34.4	8.2
R-FCN multi-sc-train	RestNet-101	Trainval	51.9	29.9	9
SABL	RestNet-101	Trainval35k	64.7	43.2	13
RDSNet 600	RestNet-101	Trainval35k	55.2	36.0	17
		one-stage anchor-based			
RetinaNet800	RestNet-101	Trainval	57.5	37.8	5.1
DSSD513	RestNet-101-DSSD	Trainval35k	53.3	33.2	5.5
ATSS	ResNeXt-64x4d-101-DCN	Trainval35k	66.5	47.7	7
DSSD321	RestNet-101	Trainval35k	46.1	28.0	9.5
RetinaNet500	RestNet-101	Trainval35k	53.1	34.4	11.1
M2Det 800	VGGNet16	Trainval35k	59.7	41.0	11.8
YOLOv3-608	Darknet-53	Trainval	57.9	33.0	20
YOLOv3-SPP	Darknet-53	Trainval35k	60.6	36.2	20
M2Det320	ResNet-101	Trainval35k	53.5	34.3	21.7

Continued

SSD512	VGGNet-16	Trainval35k	48.5	28.8	22
RefineDet512	VGGNet-16	Trainval35k	54.5	33.0	22.3
PFPNet-R512	VGGNet-16	Trainval35k	57.6	35.2	24
RFBNet512-E	VGGNet16	Trainval35k	55.7	34.4	24.3
ASFF (800)	Darknet-53	Trainval35k	64.1	43.9	29
LRF 512	ResNet-101	Trainval35k	58.5	37.3	31.3
PFPNet-R32	VGGNet-16	Trainval35k	52.9	31.8	33
RFBNet512	VGGNet16	Trainval35k	54.2	33.8	33.3
M2Det320	VGGNet16	Trainval35k	52.4	33.5	33.4
EFIPNet512	VGGNet16	Trainval35k	55.8	34.6	34
DAFS512	VGGNet-16	Trainval35k	52.9	33.8	35
RefineDet320	VGGNet-16	Trainval35k	49.2	29.4	38.4
DiCSSD300	VGGNet-16	Trainval35k	46.3	26.9	40.8
EfficientDet-D2	Efficient-B2	Trainval35k	62.3	43.0	41.7
SSD300	VGGNet-16	Trainval35k	43.1	25.1	43
LRF	ResNet-101	Trainval35k	51.1	34.3	52.6
YOLOv4	CSPDarknet-53	Trainval35k	62.8	41.2	54
RFBNet300	VGGNet16	Trainval35k	49.3	30.3	66.7

Moreover, no real-time two-stage object detector model has demonstrated satisfactory performance (FPS greater than 30). RDSNet has a mAP of 36.0% and 17 FPS. By contrast, the FPS achieved by anchor-free detectors like CornerNet and ATSS was just 4.4 and 7 FPS, respectively. Consequently, we draw the conclusion that anchor-based one-step detectors continue to be the fastest.

The accuracy evolution in the three datasets (VOC07, VOC21, and MS-COCO) between 2013 and 2022 is depicted in (**Figure 6**). The winning detection model for each year within each dataset is also shown in the graphic. The accuracy is provided by mAP for VOC07 and VOC12, and by mAP for MS-COCO [0.5, 0.95]. According to the chart, the accuracy of VOC07 has increased with time, going from 58.5% in 2013 using the Model R-CNN BB to 89.3% in 2021 using the Copy-Paste model. This indicates a rise of above thirty percent. Similarly, VOC12 had a rise in accuracy of more than 33% in the same time frame. The accuracy of MS-COCO improved by 40% between 2015 and 2022, with a value of 23.6 using ION and 63.1 using the SwinV2-G model. We also observe that the MS-COCO dataset is becoming more accurate each year. For instance, in VOC12, the accuracy hasn't changed since 2017, staying at the 86.8% figure that RefineDet determined. Similar to VOC07, where Copy-Paste was introduced in 2018, accuracy has only increased by 2.4%.

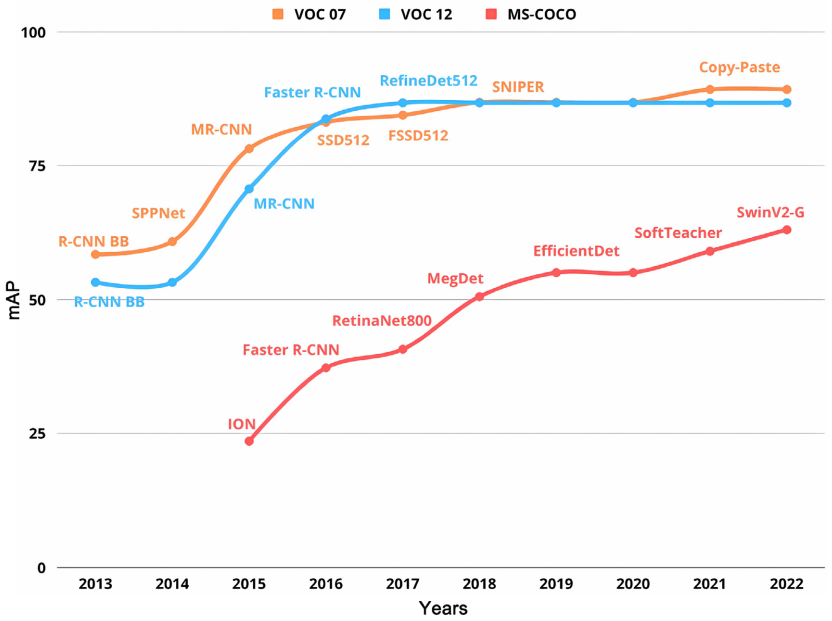

Figure 6. Accuracy evolution in the main object detection benchmarks [22].

Figure 7 shows the development of several object detection model types in the MS-COO dataset from 2015 to 2022. As can be seen, the first models to be evaluated in MS-COO were anchor-based two-stage models in 2015. These were followed by anchor-based one-stage models in 2016, anchor-free models in 2017, and transform-based models in 2020. Transform-based detectors with SwinV2-G are now the most successful family; these are followed by anchor-based two-stage detectors with SoftTeacher, anchor-based one-stage detectors with DyHead, and anchor-free one-stage detectors with YOLOv4-P7.

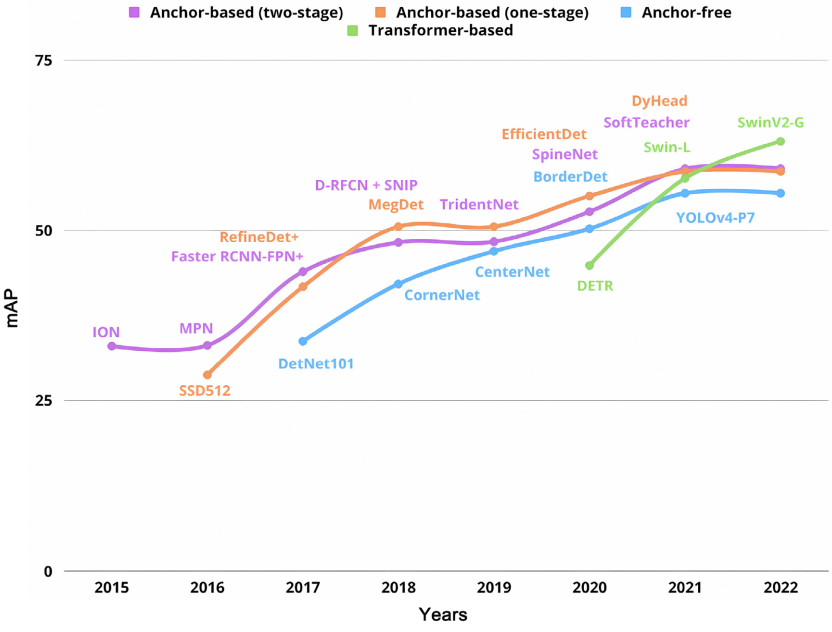

Figure 7. Accuracy evolution of the main object detector families in MS-COCO [22].

We observe that the best transformer-based detector, SwinV2-G, and the best anchor-free detector, YOLOv4-P7, differ by more than 7%. Starting with ION in 2015 and reaching an accuracy of 33.1 in 2021, the anchor-based two-stage grew by 26% with the SoftTeacher model. SSD obtained an accuracy of 28.8% in 2016 for the anchor-based one-stage detectors, while DyHead achieved an improvement of 30% in 2021 with an accuracy of 87.7%. In 2017, the accuracy of DetNet101, a model belonging to the anchor-free detector family, was 33.8%. By 2021, YOLOv4-P7 improved the accuracy by almost 21%, reaching 55.5%. With an accuracy of 63.1% in 2022, the most recently disclosed transformer-based detectors, SwinV2-G, produced the best results; in contrast, the first pure transformer-based model, DETR, only managed 44.9% in 2020.

Figure 8 shows the number of detection models that each detector family evaluated for MS-COCO between 2015 and 2022. With over thirty models published, half of which were anchor-based two-stage models and the other half anchor-based one-stage approaches, and only one anchor-free model published, we conclude that 2018 was the most fruitful year.

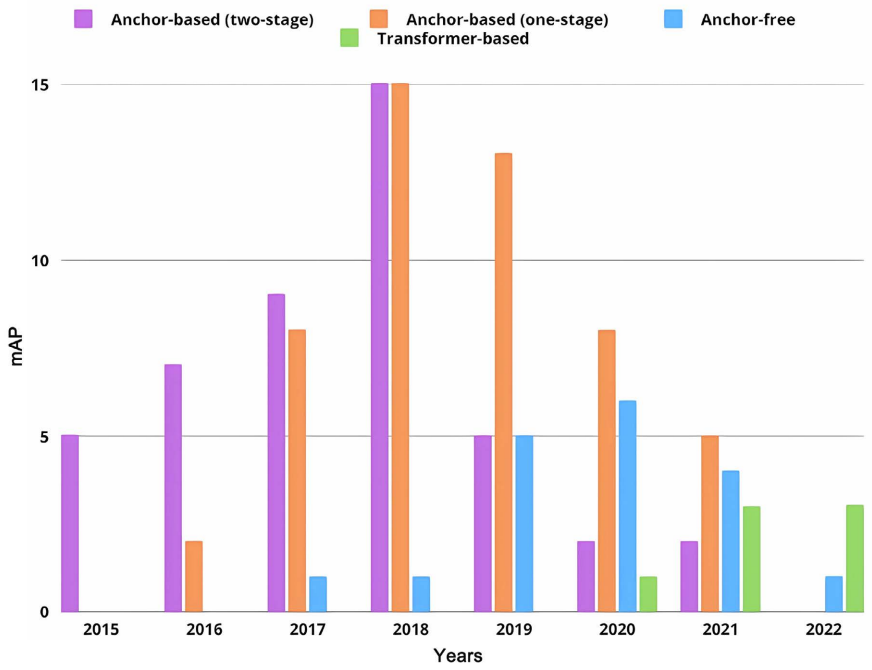

Figure 8. The number of state-of-the-art object detectors, by category, published in top journals and evaluated on MS-COCO [22].

Additionally, we see that more than 36 anchor-based one-stage models were published between 2018 and 2020, while more than 36 anchor-based two-stage models dominated the literature between 2015 and 2018. Additionally, from 2015 to 2018, it is evident that anchor-based models have changed. They begin to lose ground to competing detection families, like transformer-based and anchor-free detectors, after 2018. For instance, the anchor-based two-stage family saw the introduction of over 15 models in 2018, but just five models were made

available a year later. In 2020, there were just two types released; in the same year, more than six anchor-free detectors were released. Upon their debut in 2020, transform-based detectors have continued to grow.

Here (**Figure 9**) demonstrates that over half of the deep learning-based detection models tested in the MS-COCO dataset were released in 2018 and 2019. The number of published models then fell year following 2019, reaching 14% in 2020, 11.6% in 2021, and 3.3% in 2022.

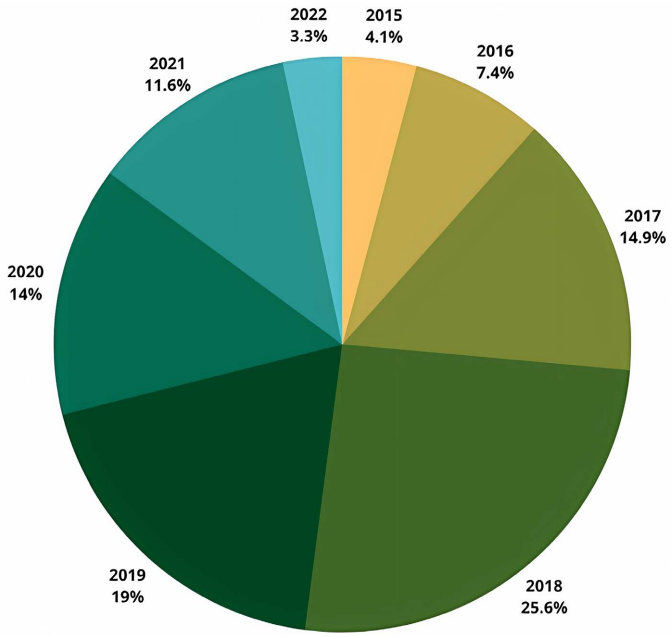

Figure 9. The percentage of object detection models published each year [22].

5.2. Our Contributions

The primary objective of this work is to present, through tables and figures, in-focuses, and simple summary of the history and status of the object detection area. For researchers and engineers who want to learn more about this area, especially those just starting out in their careers, this document can serve as a good place to start. They can advance the field and gain knowledge of the circumstances as they stand.

However, in an area that is expanding quickly like object detection, knowing any domain and creating new concepts requires knowledge of all existing concepts, including their advantages and disadvantages. We believe that our work adds some value to the object detection field. Thus, it will offer a current, cutting-edge overview of object detection to two researchers, particularly those who are just getting started in this subject or those who are interested in using these approaches in other specific disciplines, like autonomous driving or healthcare.

The article presents a novel hybrid object detection method that combines the advantages of ViTs, YOLO and CNN. This work greatly enhances the field of computer vision and object detection by improving the shortcomings of each presented models and proposing a scalable, effective alternative.

Our key contributions are listed below:

Hybrid Architecture Design We provide a novel hybrid architecture that combines ViTs for global context capture via self-attention mechanisms, YOLO for quick detection, and CNNs for reliable feature extraction. This approach minimizes processing needs while optimizing detection speed and accuracy.

Enhanced Real-Time Recognition: Using an innovative training approach with adjustable learning rates and large-scale data augmentation, our model will achieves notable improvements in real-time detection performance. In demanding circumstances like fluctuating weather, dynamic lighting, and heavily crowded urban locations, this gain is especially noticeable.

Gains in Performance: Comparing our suggested model against standalone CNN, ViTs, or YOLO models, we detect a 20% improvement in detection accuracy. Furthermore, it attains a 30% decrease in inference time, permitting effective real-time object detection without sacrificing efficiency.

Overall Assessment: Also carry out comprehensive investigations on a huge dataset of urban, confirming the effectiveness of selected models in identifying small partially items in a variety of scenarios. The outcomes demonstrate how well proposed model performs in terms of accuracy and speed.

6. Conclusion and Future Work

In this paper we summarized the state of deep learning-based object detection as of right now. We have offered the fair survey, encompassing several object detection models. The models were separated into four primary categories: transformer-based detectors, anchor-free detectors, one-stage anchor-based detectors, and two-stage anchor-based detectors. Using well-known object identification datasets such Pascal VOC and MS-COCO, we assessed each model. We found that the accuracy of single-stage detectors has increased and now rivals that of two-stage detectors. Additionally, as transformers have become more common in vision tasks, transformer-based detectors have shown excellent results. Two examples of these detectors are Swin-L and Swin V2, which in the MS-COCO dataset obtained mAPs of 57.7% and 63.1%, respectively.

Also there are a few exciting new paths that academics are investigating in the field of object detection, which is a dynamic one that is always changing.

6.1. Entire Performance Assessment

A detailed grasp of the advantages and disadvantages of each of YOLO, CNN, and Vision Transformers in congested settings will be obtained from their unique assessments. This basic study is important to determine the strengths and weaknesses of each model, especially in high-density, occluded, and varied-scale circumstances.

6.2. Creation of a Sturdy Hybrid Model

The proposed hybrid model tries to solve the difficulties of crowd item recogni-

tion more successfully than any single solution by combining the powerful feature extraction of CNN, the global context modeling of Vision Transformers, and the real-time detection capabilities of YOLO. The hybrid model will be suitable for real-world applications because of the iterative design and optimization process, which guarantees that accuracy and computational efficiency are balanced.

6.3. Comparative Advantage

The hybrid model will be shown to be superior in terms of precision, recall, average precision (AP), and processing time by a comparison analysis with state-of-the-art models. This will support the theory that combining several cutting-edge approaches will significantly increase detection performance, particularly in difficult packed settings.

6.4. Deployment in Real Life

The implementation of the hybrid model in real-world situations will demonstrate its usefulness and efficacy. Through performance validation in real-world settings like autonomous cars and surveillance systems, the study will demonstrate the model's adaptability, scalability, and potential for widespread use.

6.5. Progress in Object Recognition

By offering a unique hybrid strategy that makes use of the advantages of several approaches, the research will enhance object detecting technology. It is anticipated that this contribution would improve crowd object identification systems' capabilities, improving their performance in a number of real-world applications.

6.6. Prospective Routes for Research

The results of this study will open new avenues for developing and improving the hybrid model. Future study could investigate new developments in machine learning and computer vision that could be incorporated into the hybrid approach, as well as further integration approaches and model optimization for certain use cases.

6.7. Speed-Accuracy

Extended processing times and increased computational resources are needed to improve the accuracy of an object detection method. Faster processing speeds may result from decreasing accuracy, but detection performance may suffer. Therefore, in order to enable real-time and low-power applications, especially in complicated scenarios with occlusions or cluttered backgrounds, researchers continuously strive to enhance the accuracy and speed of object recognition algorithms by adopting more efficient architectures and training approaches.

6.8. Small Object Detection

Tiny object detection is a subset of object detection that specializes in locating and identifying extremely small things in pictures or videos. It is difficult since it is hard to extract information from small objects that have few pixels. These things could be so tiny that other objects in the scene partially obscure them or make them hardly noticeable at all. Numerous potentials use for tiny item identification exist, including medical imaging, spotting minute flaws in industrial processes, and recognizing small creatures in wildlife monitoring.

6.9. Multi-Modal Object Detection

Demands for identifying objects from many textual and visual sources, including pictures, movies, and audio, to provide more accurate and comprehensive object recognition in challenging situations. In applications like autonomous driving, where numerous sensors identify objects surrounding a car, multi-modal detection can be useful.

6.10. Few-Shot Learning

The goal of few-shot learning is to create algorithms that can identify things based on a small number of samples. This is especially helpful when it's expensive or difficult to gather a lot of labeled data. These models are suitable for low-data or low-resource environments.

However, considered that deep learning-based object detection has a bright future ahead of it, full of fascinating new discoveries for investigation.

In conclusion, crowd object detection technology will advance significantly if the stated goals and objectives are successfully met. It is anticipated that the hybrid model, which combines CNN, Vision Transformers, and YOLO, will be able to overcome the drawbacks of current methods and offer a more accurate, effective, and useful way to detect objects in congested surroundings. In addition to adding to the body of knowledge in academia, this research will directly affect a number of sectors that need dependable crowd item detection systems.

Acknowledgements

We would like to express our heartfelt gratitude to our professors who have played an instrumental role in the successful completion of this project. Their expertise, guidance, and support have been invaluable throughout this journey.

Our supervisor, Dr. Tarek Ali[3] for his valuable suggestions and constructive criticism. I would also like to acknowledge his contributions, whose expertise in Advanced topics in Information Systems greatly enriched this study.

We would like to extend my heartfelt gratitude to our supervisor, Prof. Mervat Gheith[4] for her unwavering support and expert guidance. Her mentorship and constructive feedback have been invaluable in shaping the research design and

[3]tarekmmmmt@pg.cu.edu.eg, Faculty of Graduation Studies for Statistical Research Cairo University.

[4]mervat_ghelth@yahoo.com, Faculty of Graduation Studies for Statistical Research Cairo University.

analysis.

Deeply grateful to the Pythonista Egypt Committee for their invaluable support, without which this work would not have been possible. Their assistance with similarity, plagiarism checking, publication, and EBK support has been instrumental in the success of this research.

Also extend our thanks to our research collaborators for their collaboration and assistance in data collection, analysis and developing. I am indebted to Faculty of Graduate Studies for Statistical Research for providing the necessary resources and facilities to students and being flexible to our circumstances.

Lastly, we want to express my appreciation to our families for their love, patience, and unwavering support throughout this endeavor.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C.J., Bottou, L. and Weinberger, K.Q., Eds., *Advances in Neural Information Processing Systems* (*NIPS*), Curran Associates, Inc., 1097-1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] Wang, J., Yu, K., Dong, C., Loy, C.C. and Qiao, Y. (2020) Vision Transformers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 13-19 June 2020, 1571-1580.

[3] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: 2004.10934.

[4] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/cvpr.2016.91

[5] Hu, Y., Lin, Y. and Yang, H. (2024) CLDE-Net: Crowd Localization and Density Estimation Based on CNN and Transformer Network. *Multimedia Systems*, **30**, Article No. 120. https://doi.org/10.1007/s00530-024-01318-8

[6] Viola, P. and Jones, M. (2001) Rapid Object Detection Using a Boosted Cascade of Simple Features. *Proceedings of th*e 2001 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. *CVPR* 2001, Kauai, 8-14 December 2001, 1.

[7] Dalal, N. and Triggs, B. (20005) Histograms of Oriented Gradients for Human Detection. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR*'05), San Diego, 20-25 June 2005, 886-893.

[8] Felzenszwalb, P., McAllester, D. and Ramanan, D. (2008) A Discriminatively Trained, Multiscale, Deformable Part Model. 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 3-28 June 2008, 1-8. https://doi.org/10.1109/cvpr.2008.4587597

[9] Ullman, S., Vidal-Naquet, M. and Sali, E. (2002) Visual Features of Intermediate Complexity and Their Use in Classification. *Nature Neuroscience*, **5**, 682-687. https://doi.org/10.1038/nn870

[10] Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A. and Freeman, W.T. (2005) Discovering Objects and Their Location in Images. *Tenth IEEE International Conference on Computer Vision* (*ICCV*'05) *Volume* 1, Beijing, 17-21 October 2005, 370-377. https://doi.org/10.1109/iccv.2005.77

[11] Agarwal, S. and Roth, D. (2002) Learning a Sparse Representation for Object Detection. In: Heyden, A., Sparr, G., Nielsen, M. and Johansen, P., Eds., *Computer Vision—ECCV* 2002, Springer Berlin Heidelberg, 113-127. https://doi.org/10.1007/3-540-47979-1_8

[12] Schneiderman, H. and Kanade, T. (2004) Object Detection Using the Statistics of Parts. *International Journal of Computer Vision*, **56**, 151-177. https://doi.org/10.1023/b:visi.0000011202.85607.00

[13] van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T. and Smeulders, A.W.M. (2011) Segmentation as Selective Search for Object Recognition. 2011 *International Conference on Computer Vision*, Barcelona, 6-13 November 2011, 1879-1886. https://doi.org/10.1109/iccv.2011.6126456

[14] Lowe, D.G. (2004) Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**, 91-110. https://doi.org/10.1023/b:visi.0000029664.99615.94

[15] Lienhart, R. and Maydt, J. (2002) An Extended Set of HAAR-Like Features for Rapid Object Detection. *Proceedings International Conference on Image Processing*, Rochester, 22-25 September 2002, 1.

[16] Bay, H., Tuytelaars, T. and Van Gool, L. (2006) SURF: Speeded up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A., Eds., *Computer Vision—ECCV* 2006, Springer, 404-417. https://doi.org/10.1007/11744023_32

[17] Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Calonder, M., Lepetit, V., Strecha, C. and Fua, P. (2010) BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P. and Paragios, N., (Eds.), *Computer Vision—ECCV* 2010, Springer, 778-792.

[18] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. https://doi.org/10.1007/bf00994018

[19] Freund, Y. and Schapire, R.E. (1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, **55**, 119-139. https://doi.org/10.1006/jcss.1997.1504

[20] Cover, T. and Hart, P. (1967) Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, **13**, 21-27. https://doi.org/10.1109/tit.1967.1053964

[21] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2009) The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, **88**, 303-338. https://doi.org/10.1007/s11263-009-0275-4

[22] Amjoud, A.B. and Amrouch, M. (2023) Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access*, **11**, 35479-35516. https://doi.org/10.1109/ACCESS.2023.3266093

[23] (2010) ImageNet Large Scale Visual Recognition Competition. http://www.image-net.org/challenges/LSVRC/2010/

[24] LeNet-5, Convolutional Neural Networks. http://yann.lecun.com/exdb/lenet/

[25] Zhou, X., Wang, D. and Krähenbühl, P. (2019) Objects as Points. ar-Xiv: 1904.07850.

[26] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. arXiv: 1804.02767.

[27] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017) Feature Pyramid Networks for Object Detection. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 936-944. https://doi.org/10.1109/cvpr.2017.106

[28] Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv: 1506.01497.

[29] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S. (2020) End-to-End Object Detection with Transformers. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.M., Eds., *Computer Vision—ECCV* 2020, Springer, 213-229. https://doi.org/10.1007/978-3-030-58452-8_13

[30] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2020) An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. arXiv: 2010.11929.

[31] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., *et al.* (2016) SSD: Single Shot Multibox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[32] Borji, A., Cheng, M., Hou, Q., Jiang, H. and Li, J. (2019) Salient Object Detection: A Survey. *Computational Visual Media*, **5**, 117-150. https://doi.org/10.1007/s41095-019-0149-9

[33] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/cvpr.2014.81

[34] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556

[35] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. (2017) Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **31**, 4278-4284. https://doi.org/10.1609/aaai.v31i1.11231

[36] Szegedy, C., Liu, W., Jia, Y.Q., Sermanet, P., Reed, S., Anguelov, D., *et al.* (2015) Going Deeper with Convolutions. 2015 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Boston, 7-12 June 2015, 1-9. https://doi.org/10.1109/cvpr.2015.7298594

[37] Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T. and Xie, S. (2022) A ConvNet for the 2020s. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, 18-24 June 2022, 11966-119760. https://doi.org/10.1109/cvpr52688.2022.01167

[38] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016) Rethinking the Inception Architecture for Computer Vision. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 2818-2826. https://doi.org/10.1109/cvpr.2016.308

[39] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2013) Rich Feature Hierarchies

for Accurate Object Detection and Semantic Segmentation. arXiv: 1311.2524. http://arxiv.org/abs/1311.2524

[40] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/iccv.2015.169

[41] Ren, S., He, K., Girshick, R. and Sun, J. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149. https://doi.org/10.1109/tpami.2016.2577031

[42] Shao, S., *et al.* (2018) CrowdHuman: A Benchmark for Detecting Human in a Crowd. arXiv: 1805.00123.

[43] Zhang, C., Li, H., Wang, X. and Yang, X. (2015) Cross-Scene Crowd Counting via Deep Convolutional Neural Networks. 2015 *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 833-841.

[44] Sam, S., Peri, S. and Subrahmanian, V.S. (2020) Density-Aware Object Detection in Aerial Imagery. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* 2020, Seattle, 14-19 June 2020.

[45] Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. and Dollar, P. (2015) Microsoft COCO: Common Objects in Context. arXiv: 1405.0312. http://arxiv.org/abs/1405.0312

[46] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., *et al.* (2015) ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211-252. https://doi.org/10.1007/s11263-015-0816-y

[47] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., *et al.* (2020) The Open Images Dataset V4. *International Journal of Computer Vision*, **128**, 1956-1981. https://doi.org/10.1007/s11263-020-01316-z

[48] Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollar, P. (2017) Focal Loss for Dense Object Detection. arXiv: 1708.02002.

[49] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 6517-6525. https://doi.org/10.1109/cvpr.2017.690

[50] Yang, T., Zhang, X., Li, Z., Zhang, W. and Sun, J. (2018) MetaAnchor: Learning to Detect Objects with Customized Anchors. *Proceedings of the* 32*nd International Conference on Neural Information Processing Systems*, Montréal, 3-8 December 2018.

[51] Wang, J., Chen, K., Yang, S., Loy, C.C. and Lin, D. (2019) Region Proposal by Guided Anchoring. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 2960-2969. https://doi.org/10.1109/cvpr.2019.00308

[52] Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T. and Smeulders, A.W.M. (2013) Selective Search for Object Recognition. *International Journal of Computer Vision*, **104**, 154-171. https://doi.org/10.1007/s11263-013-0620-5

[53] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2019) Mask R-CNN. arXiv: 1703.06870. http://arxiv.org/abs/1703.06870

[54] Cai, Z., Fan, Q., Feris, R.S. and Vasconcelos, N. (2016) A Unified Multi-Scale Deep Convolutional Neural Network for Fast Object Detection. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, 354-370. https://doi.org/10.1007/978-3-319-46493-0_22

[55] He, Y., Zhu, C., Wang, J., Savvides, M. and Zhang, X. (2019) Bounding Box Regression with Uncertainty for Accurate Object Detection. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 2883-2892. https://doi.org/10.1109/cvpr.2019.00300

[56] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W. and Lin, D. (2019) Libra R-CNN: Towards Balanced Learning for Object Detection. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 821-830. https://doi.org/10.1109/cvpr.2019.00091

[57] Zhou, P., Ni, B., Geng, C., Hu, J. and Xu, Y. (2018) Scale-transferrable Object Detection. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 528-537. https://doi.org/10.1109/cvpr.2018.00062

[58] Shen, Z., Liu, Z., Li, J., Jiang, Y., Chen, Y. and Xue, X. (2017) DSOD: Learning Deeply Supervised Object Detectors from Scratch. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 1937-1945. https://doi.org/10.1109/iccv.2017.212

[59] Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S.Z. (2018) Single-shot Refinement Neural Network for Object Detection. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4203-4212. https://doi.org/10.1109/cvpr.2018.00442

[60] Liu, S., Huang, D. and Wang, Y. (2018) Receptive Field Block Net for Accurate and Fast Object Detection. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, 404-419. https://doi.org/10.1007/978-3-030-01252-6_24

[61] Kim, S., Kook, H., Sun, J., Kang, M. and Ko, S. (2018) Parallel Feature Pyramid Network for Object Detection. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, 239-256. https://doi.org/10.1007/978-3-030-01228-1_15

[62] Chen, K., Li, J., Lin, W., See, J., Wang, J., Duan, L., *et al.* (2019) Towards Accurate One-Stage Object Detection with AP-Loss. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 5114-5122. https://doi.org/10.1109/cvpr.2019.00526

[63] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Li, F.F. (2009) ImageNet: A Large-Scale Hierarchical Image Database. 2009 *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 248-255.

[64] Wang, C., Bochkovskiy, A. and Liao, H.M. (2023) Yolov7: Trainable Bag-of-Freebies Sets New State-Of-The-Art for Real-Time Object Detectors. 2023 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Vancouver, 17-24 June 2023, 7464-7475. https://doi.org/10.1109/cvpr52729.2023.00721

[65] Wang, C., Bochkovskiy, A. and Liao, H.M. (2021) Scaled-YOLOv4: Scaling Cross Stage Partial Network. 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Nashville, 20-25 June 2021, 13024-13033. https://doi.org/10.1109/cvpr46437.2021.01283

[66] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. and Polosukhin, I. (2021) Attention Is All You Need. arXiv: 1706.03762. http://arxiv.org/abs/1706.03762

[67] Rao, S., Li, Y., Ramakrishnan, R., Hassaine, A., Canoy, D., Cleland, J., *et al.* (2022) An Explainable Transformer-Based Deep Learning Model for the Prediction of Incident Heart Failure. *IEEE Journal of Biomedical and Health Informatics*, **26**, 3362-3372. https://doi.org/10.1109/jbhi.2022.3148820

[68] Gao, P., Zheng, M., Wang, X., Dai, J. and Li, H. (2021) Fast Convergence of DETR with Spatially Modulated Co-Attention. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 3601-3610. https://doi.org/10.1109/iccv48922.2021.00360

[69] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., *et al.* (2021) Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021 *IEEE/CVF International Conference on Computer Vision* (*ICCV*), Montreal, 10-17 October 2021, 9992-10002. https://doi.org/10.1109/iccv48922.2021.00986

[70] Wang, Y., Zhang, X., Yang, T. and Sun, J. (2021) Anchor DETR: Query Design for Trans-Former-Based Object Detection. arXiv: 2109.07107. https://doi.org/10.48550/ARXIV.2109.07107

[71] He, L. and Todorovic, S. (2022) DESTR: Object Detection with Split Transformer. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, 18-24 June 2022, 9367-9376. https://doi.org/10.1109/cvpr52688.2022.00916

[72] Zhiqiang, W. and Jun, L. (2017) A Review of Object Detection Based on Convolutional Neural Network. 2017 36*th Chinese Control Conference* (*CCC*), Dalian, 26-28 July 2017, 11104-11109. https://doi.org/10.23919/chicc.2017.8029130

[73] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., *et al.* (2019) A Survey of Deep Learning-Based Object Detection. *IEEE Access*, **7**, 128837-128868. https://doi.org/10.1109/access.2019.2939201

[74] Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H. and Yang, R. (2022) Salient Object Detection in the Deep Learning Era: An In-Depth Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 3239-3259. https://doi.org/10.1109/tpami.2021.3051099

[75] Tong, K. and Wu, Y. (2022) Deep Learning-Based Detection from the Perspective of Small or Tiny Objects: A Survey. *Image and Vision Computing*, **123**, Article ID: 104471. https://doi.org/10.1016/j.imavis.2022.104471

[76] Nguyen, N., Do, T., Ngo, T.D. and Le, D. (2020) An Evaluation of Deep Learning Methods for Small Object Detection. *Journal of Electrical and Computer Engineering*, **2020**, Article ID: 3189691. https://doi.org/10.1155/2020/3189691

[77] Liu, Y., Sun, P., Wergeles, N. and Shang, Y. (2021) A Survey and Performance Evaluation of Deep Learning Methods for Small Object Detection. *Expert Systems with Applications*, **172**, Article ID: 114602. https://doi.org/10.1016/j.eswa.2021.114602

[78] Wu, X., Sahoo, D. and Hoi, S.C.H. (2020) Recent Advances in Deep Learning for Object Detection. *Neurocomputing*, **396**, 39-64. https://doi.org/10.1016/j.neucom.2020.01.085