Scripta

Winter 2024, Vol. 3, No. 2, 1-23
ISSN: 2769-9242

Object Detection Models: Convolutional Neural Networks vs. Vision Transformers

Arnav Arora* and Ramviyas Parasuraman †

Abstract

This paper examines the role of Convolutional Neural Networks (CNNs) and vision transformers in the context of object detection, a fundamental task in computer vision. Beginning with an overview encompassing the definition and significance of object detection, the paper traces the evolution from traditional methods relying on handcrafted features to the revolutionary impact of deep learning architectures. CNNs, inspired by the hierarchical organization of the visual cortex, emerged as a powerful tool for object detection, with landmark architectures such as AlexNet, VGGNet, GoogLeNet, and ResNet pushing the boundaries of performance and scalability. Each architecture introduced unique innovations, leading to significant improvements in object detection accuracy and speed. In parallel, the rise of vision transformers marked a paradigm shift in computer vision. Inspired by the success of transformers in natural language processing, vision transformers propose a novel architecture that utilizes self-attention mechanisms to process visual data. Breaking away from the sole pixel-wise processing of CNNs, vision transformers demonstrate impressive performance on image classification tasks and show promise for various visual recognition tasks, including object detection. Challenges such as real-time processing, scalability, and robustness are discussed in this paper, along with future prospects such as Liquid Time-constant Networks.

Introduction

"... we will be using the term object recognition broadly to encompass both image classification (a task requiring an algorithm to determine what object classes are present in the image) as well as object detection (a task requiring an algorithm to localize all objects present in the image" — ImageNet Large Scale Visual Recognition Challenge, 2015. [1].

The primary goal of object detection is to accurately detect and classify objects of interest within a scene. This is typically achieved by drawing bounding boxes around the detected objects and assigning them to predefined categories. Early methods for object detection relied on handcrafted features and classifiers, such as Haar cascades and Histogram of Oriented

*College Freshman, Georgia Tech.
Contact: aarora338@gatech.edu
† Researcher, University of Georgia.
Contact: ramviyas@uga.edu

Gradients (HOG) [2]. The introduction of deep learning, particularly convolutional neural networks (CNNs), revolutionized object detection. Models like R-CNN (Region-based Convolutional Neural Networks), Fast R-CNN [3], and Faster R-CNN significantly improved detection accuracy and speed. Single-shot detection (SSD) [4] and You Only Look Once (YOLO) [5] architectures further improved speed by performing detection in a single pass through the network.

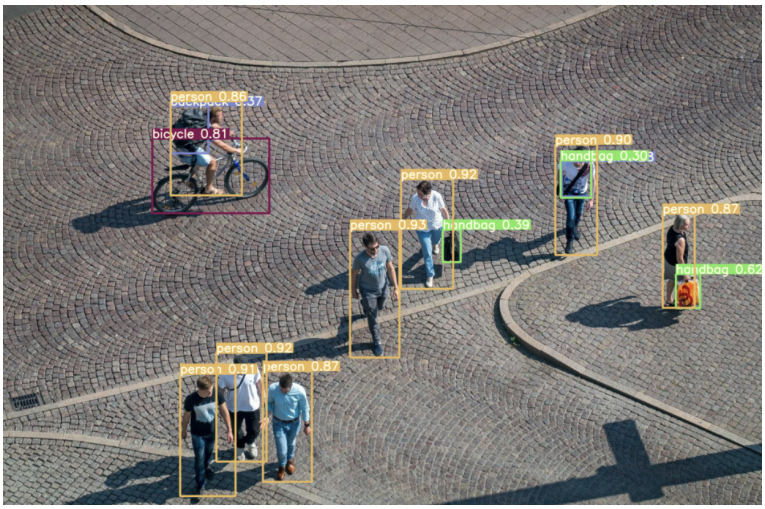

Figure 1: Trained Model Yolo v8 is capable of object detection on live videos [6]

Vision transformers, inspired by the success of transformers in Natural Language Processing (NLP), propose a fundamentally different architecture for processing visual data. The key innovation introduced by vision transformers is the application of self-attention mechanisms to process image patches. In contrast to CNNs, which process the entire image pixel by pixel, vision transformers break down the input image into smaller, fixed-size patches.[7] ViT demonstrated impressive performance on image classification tasks, rivaling state-of-the-art CNN architectures such as ResNet [8] and EfficientNet[9]. The emergence of vision transformers represents a new path in the world of computer vision, challenging the dominance of CNNs and paving the way for new architectures and methodologies. While vision transformers are relatively newer compared to CNN's, their potential to revolutionize a wide range of visual recognition tasks, including object detection, holds promise for the future of computer vision.

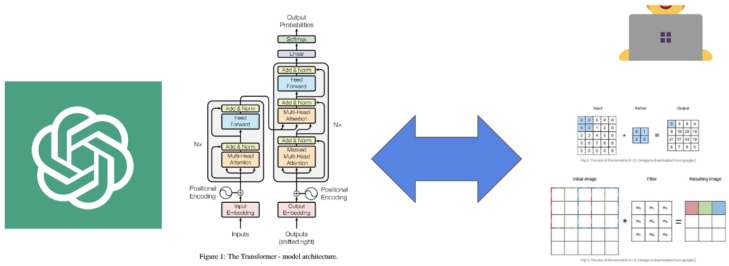

Figure 2: Illustrative comparison between CNN architecture and vision transformer architecture

The accuracy of an object detection model hinges on several factors, including training sample quality, input imagery, model parameters, and accuracy thresholds. Object detection can be assessed through various metrics, including precision, recall, F1 score, precision-recall curve, Average Precision (AP), Mean Average Precision (mAP), and the Intersection over Union (IoU) ratio [10].

Metrics for Object Detection

The formulas for such metrics is as follows:

Precision is a measurement for how accurate predictions are. It is calculated through the formula

$$Precision = \frac{TP}{TP + FP}$$

with TP as true positive and FP as false positive predictions. In object detection, precision indicates the proportion of correctly identified objects among all objects predicted by the model. [11]

Intersection over union (IoU) gives the ratio of overlap between two boxes. Measures the overlap between predicted and ground-truth bounding boxes. With bg being the ground truth bounding box and bp being the predicted bounding box, the IoU of these boxes can be calculated with

$$IoU = \frac{area\_of\_intersection}{area\_of\_union} = \frac{A(bg \cap bp)}{A(bg \cup bp)}$$

The **precision-recall curve** is a graphical representation of the trade-off between precision and recall at various threshold settings. It provides

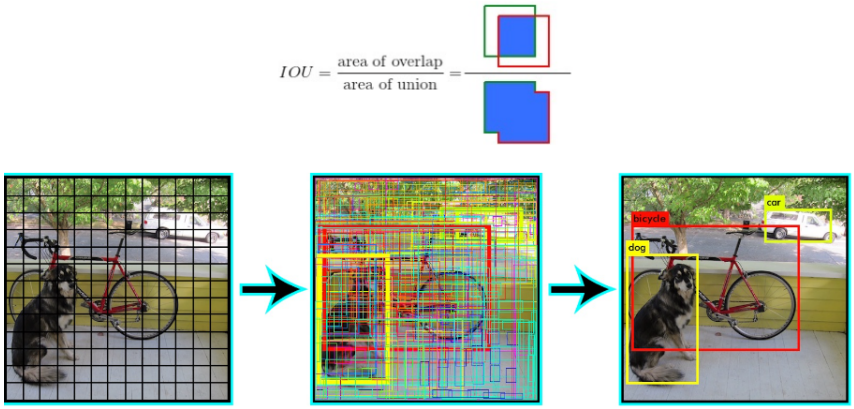

Figure 3: Non-Maximum Suppression with IOU Equation for detecting [12] bounding boxes on image

valuable insights into the model's performance across different operating points, where precision is on the y-axis against recall on the x-axis, with each point representing a different threshold setting.

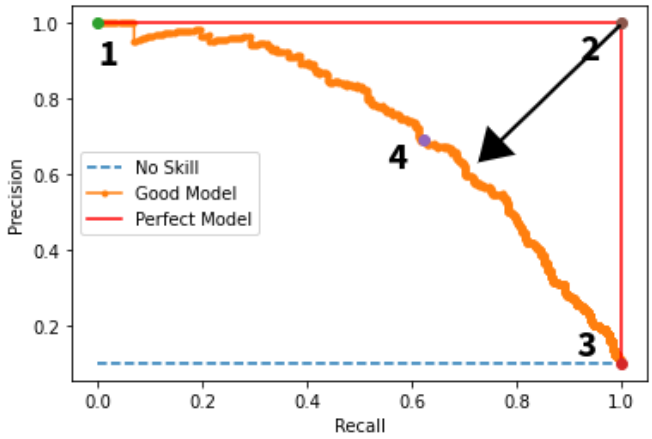

Figure 4: On the curves, each point corresponds to a different threshold and its Precision and Recall [13]

Mean Average Precision (mAP) is a widely used metric in object detection tasks to evaluate the overall performance of a model across multiple classes. It provides a comprehensive assessment of the model's performance across various object categories. A higher mAP value indicates better overall performance of the model across all classes. The mAP

formula is expressed as:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

And is one of the main metrics that will be used to analyze the efficiency and accuracy of the models we are working with.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have played a pivotal role in advancing the field of object detection. Their hierarchical architecture, inspired by the visual cortex's organization in the human brain, enables effective feature extraction and hierarchical representation learning from raw image data.

Role and History

CNNs ability to automatically learn hierarchical features from images has propelled them to the forefront of visual recognition tasks. The history of CNNs in computer vision can be traced back to the pioneering work of LeCun with the LeNet-5 architecture in the early 1990s. The network has 5 layers with learnable parameters, it has 3 convolution layers, two average pooling layers, and two fully connected layers with a softmax classifier.

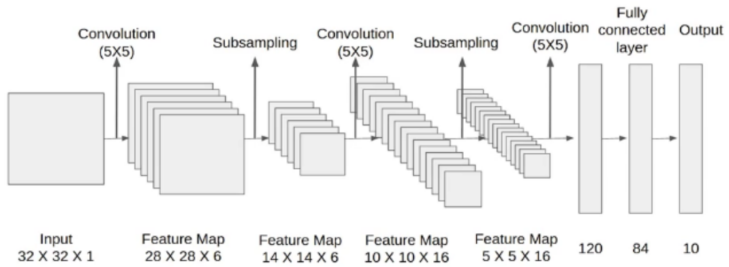

Figure 5: The main reason behind the popularity of this model was its straightforward architecture. Figure reproduced from [14] for demonstration of LeNet-5 architecture

However, it was not until the breakthrough success of Krizhevsky with the AlexNet architecture in 2012 that CNNs gained widespread attention

and adoption. AlexNet, with its deep architecture comprising multiple convolutional and pooling layers, achieved remarkable performance (for its time) on the COCO dataset, as well as winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a significant margin. Following the success of AlexNet, numerous CNN architectures have been proposed, each pushing the boundaries of performance and scalability in object detection:

VGGNet: Introduced by Simonyan and Zisserman in 2014, VGGNet featured a uniform architecture with small 3x3 convolutional filters stacked on top of each other. Despite its simplicity, VGGNet achieved competitive performance on various image classification tasks. [15]

GoogLeNet (Inception): Developed by Szegedy and others at Google in 2014, GoogLeNet introduced the concept of inception modules, which allowed for efficient utilization of computational resources by performing convolutions at multiple scales within the same layer. [16]

ResNet (Residual Networks): Proposed by Kaiming He and Zhangin in 2015, ResNet introduced residual connections that enabled the training of very deep neural networks. This architecture mitigated the vanishing gradient problem, allowing for the successful training of CNNs with hundreds or even thousands of layers. [17]

These landmark CNN architectures laid the foundation for major advancements in object detection. They demonstrated the effectiveness of deep learning techniques in learning hierarchical representations directly from raw image data.

Formulation of CNN

A CNN's architecture comprises of convolutional layers, pooling layers, and fully connected layers, collectively designed to extract and learn intricate patterns from raw image data.

In the formulation of CNNs, convolutional layers serve as the backbone for feature extraction, convolving learnable filters across the input image to detect local patterns and features.

Mathematically, the convolution operation can be represented as:

$$(f * g)(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} f(i, j) \cdot g(x - i, y - j)$$

where:

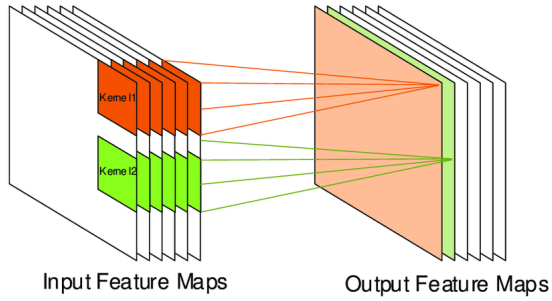

Figure 6: Operation of a convolutional layer, with input feature maps, convolutional filters, and output feature maps

- f is the input image,
- g is the convolutional filter/kernel,
- (x, y) are the coordinates of the output feature map,
- (i, j) are the coordinates within the filter/kernel,
- m and n are the dimensions of the filter/kernel.

These convolutional operations are coupled with non-linear activation functions, such as ReLU, to introduce non-linearity and enable the network to learn complex relationships within the data.

The Rectified Linear Unit (ReLU) is a commonly used activation function, defined as:

$$f(x) = \max(0, x)$$

where x represents the input to the activation function.

Pooling layers, typically implemented as max-pooling or average-pooling operations, follow convolutional layers to downsample feature maps, reducing spatial dimensions while preserving essential information. This downsampling process enhances the network's ability to extract robust and invariant features, thereby facilitating translation invariance and reducing computational complexity.

Max-pooling is one of the most common pooling operations, and it can be represented mathematically as:

$$\text{MaxPooling}(x, y) = \max_{i=1}^{k} \max_{j=1}^{k} f(x + i, y + j)$$

where k is the size of the pooling window

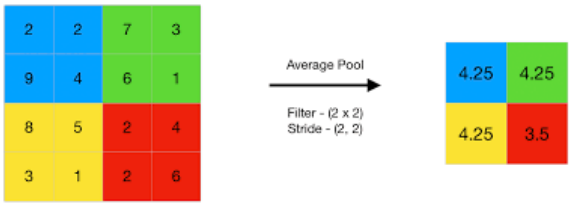

Figure 7: pooling layers downsample feature maps to reduce spatial dimensions while preserving essential information

Finally, fully connected layers, situated at the end of the network, integrate the extracted features to make predictions or classifications. These layers leverage the hierarchical representations learned by earlier convolutional and pooling layers to capture high-level abstractions and make informed decisions.

In the fully connected layer, the output from the previous layer is flattened into a vector and multiplied by a weight matrix W, followed by the addition of a bias vector b. Mathematically, this operation can be expressed as:

$$\mathrm{FC}(x) = Wx + b$$

where:

- x is the input vector,
- W is the weight matrix,
- b is the bias vector.

CNN Models on COCO Validation Set

Several Convolutional Neural Network models have been evaluated on the COCO validation set. Metrics were collected but for the sake of simplicity, only the mAP50 and inference times are mentioned. Some models include:

- **Faster R-CNN**: Faster R-CNN is a two-stage object detection model proposed by Ren et al. [3]. It consists of a Region Proposal Network (RPN) followed by a Region-based Convolutional Neural Network (RCNN). Faster R-CNN achieves high accuracy by generating region proposals and then classifying objects within these proposals with a mAP50 of 59.1% and inference time 172 ms

- **SSD (Single Shot MultiBox Detector)**: SSD is another single-stage object detection model introduced by Liu et al. [4]. It predicts object bounding boxes and class probabilities directly from feature maps at multiple scales, enabling efficient and accurate detection. For SSD321 (input 321×321), the mAP50 score was 45.4% on average with an inference time of 61 ms.

- **RetinaNet**: RetinaNet is a single-stage object detection model that addresses the problem of class imbalance inherent in object detection datasets. It utilizes a focal loss function to down-weight easy examples and focus training on hard examples. With the RetinaNet-101-500 model, the mAP50 score was 53.1% and inference time was 90 ms

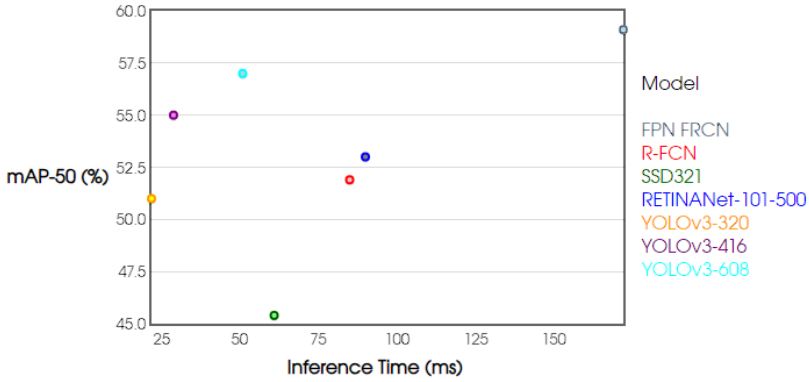

Figure 8: Results Plotted

From the pioneering work of LeCun with LeNet-5[18] to the breakthrough success of models like Faster R-CNN, YOLO, SSD, and RetinaNet[19], the evolution of CNNs has reshaped how we perceive and interact with visual data. These models have not only achieved remarkable performance on benchmark datasets like COCO but have also paved the way for real-world applications in autonomous vehicles, surveillance systems, and medical imaging, among others. As research continues to push the boundaries of object detection, leveraging the strengths of CNNs remains crucial in developing robust and efficient solutions for a wide range of visual recognition tasks. Let's explore how the concept of Transformers, originally introduced in natural language processing, has been adapted and applied to computer vision tasks, opening up new avenues for innovation

and exploration.

Vision Transformers

The concept of Transformers was first introduced in the seminal paper "Attention is All You Need" by Vaswani et al. [20], which revolutionized NLP by leveraging self-attention mechanisms to capture long-range dependencies in sequences of tokens. Building upon this success, researchers explored the application of transformer architectures to computer vision tasks, giving rise to Vision Transformers.

Formulation of Vision Transformer

The transformer architecture is increasingly being explored as an alternative to convolutional neural networks (CNNs) in computer vision tasks. Vision transformers employed as backbone networks for image classification, replacing the final stage of convolutions in the ResNet architecture, utilize convolutional layers to extract low-level features. The vision transformer employs a tokenizer to group pixels into visual tokens, each representing a semantic concept within the image. These visual tokens are directly used for image classification, while the transformer captures relationships between the tokens. This method allows for the modeling of complex image structures and has shown promising results in various visual recognition tasks.

The architecture typically consists of an initial tokenization layer, followed by multiple transformer encoder layers. Each transformer encoder layer comprises self-attention modules, feedforward neural networks, and residual connections, facilitating effective feature extraction and representation learning from visual data.

Tokenization Layer

The tokenization layer of a Vision Transformer converts the input image into a sequence of tokens, each representing an image patch. Mathematically, the tokenization process can be represented as follows:

$$\text{Tokens} = \text{Split\_Image}(I)$$

where I is the input image and Split_Image denotes the function that splits the image into patches.

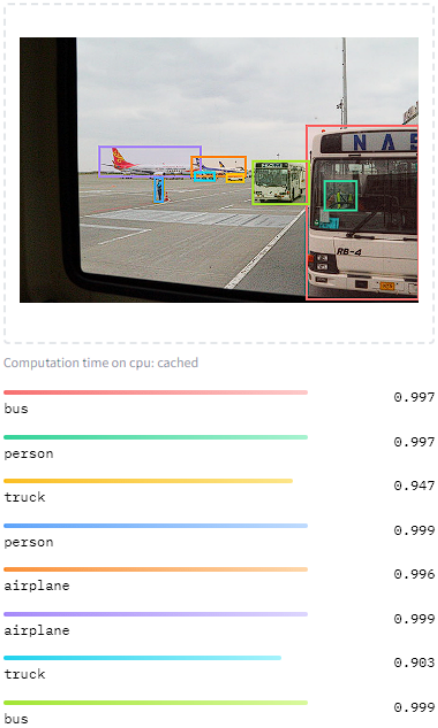

Figure 9: Object Detection with DETR Model [21]

Positional Encoding

In addition to the image patches, ViTs require positional information to capture spatial relationships within the image. Positional encoding is typically added to the token embeddings to provide this information. The positional encoding is calculated using sinusoidal functions to encode the position of each token along different dimensions.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$
$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

where pos represents the position of the token, i represents the dimension of the positional encoding, and d represents the dimensionality of the token embeddings.

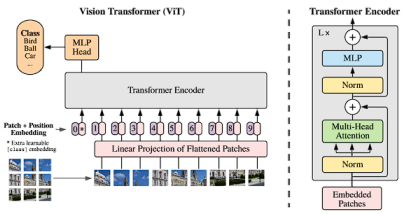

Figure 10: The framework of a Vision Transformer [22]

Transformer Encoder Layers

The core of the Vision Transformer architecture consists of multiple transformer encoder layers, each comprising self-attention mechanisms and feedforward neural networks.

The self-attention mechanism calculates a weighted sum of the input embeddings, allowing each token to attend to relevant tokens in the sequence.[23] Mathematically, the self-attention mechanism can be expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k represents the dimensionality of the key vectors.

The incorporation of residual connections facilitates the flow of information through the network, mitigating the vanishing gradient problem and enabling the training of deeper architectures.

The output of the self-attention mechanism is then passed through a feedforward neural network, which applies a series of linear transformations followed by activation functions such as ReLU. [22] The feedforward network can be represented as follows:

$$\text{FFN}(x) = \text{ReLU}(W_1 x + b_1)W_2 + b_2$$

where W_1, b_1, W_2, and b_2 represent the weight matrices and bias vectors of the feedforward network.

ViT Models on COCO Validation Set

Several models incorporating Vision Transformers (backbone or attention encoding) have been evaluated on the COCO validation set, showcasing

the effectiveness of ViT concepts in object detection tasks. Metrics were collected but for the sake of simplicity, only the mAP50 and inference times are mentioned. Some models include:

- **YOLOv3**: YOLOv3, short for You Only Look Once version 3, is a popular object detection model known for its real-time performance[24]. YOLOv3 is renowned for its exceptional speed and accuracy in object detection. Achieving comparable mean Average Precision (mAP) to models using Focal Loss but operating around four times faster, YOLOv3 offers a unique advantage in adjusting between speed and accuracy by simply modifying the model size, eliminating the need for retraining. Unlike previous detection systems, YOLOv3 applies a single neural network to the entire image, dividing it into regions and predicting bounding boxes and probabilities for each region. This approach allows YOLOv3 to benefit from global context during testing, enhancing accuracy, and making predictions with a single network evaluation, resulting in exceptional speed [25]. YOLOv3 incorporates various optimizations such as multi-scale predictions, a refined backbone classifier, and more, further enhancing its performance in object detection tasks. Received a mAP50 of 55.3 with an inference time of 29 ms for YOLOv3-416.

- **DETR with SWIN-L Backbone**: Detection Transformer (DETR) is a groundbreaking object detection model that predicts all objects simultaneously. It is trained end-to-end with a set loss function, which performs bipartite matching between predicted and ground-truth objects, thereby simplifying the detection pipeline. DETR eliminates the need for multiple hand-designed components that encode prior knowledge, such as spatial anchors or non-maximal suppression. When combined with the SWIN-L backbone, DETR leverages the hierarchical processing capabilities of the SWIN transformer, demonstrating competitive performance on various object detection benchmarks. Compared to most previous work on direct set prediction, the main features of DETR are the conjunction of the bipartite matching loss and transformers with parallel decoding. This is in contrast to previous work that focused on autoregressive decoding with RNNs. DETR's matching loss function uniquely assigns a prediction to a ground truth object and is invariant to a permutation of predicted objects, allowing us to emit them in parallel. This highlights the

potential of DETR with a SWIN-L backbone in advancing object detection methodologies. This model received a mAP50 of 68.4% with an inference time of 40ms on average.

- **RT-DETR-L (Real-Time Detection Transformer)**: RT-DETR-L is an advanced variant of the Detection Transformer (DETR) architecture tailored for real-time object detection tasks. It comprises a sophisticated backbone, an efficient hybrid encoder, and a Transformer decoder equipped with auxiliary prediction heads.

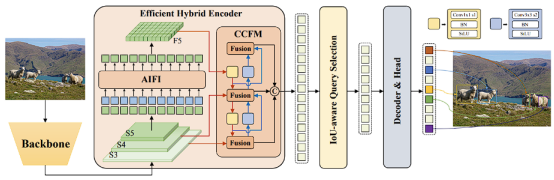

Figure 11: Overview of RT-DETR [26]

The backbone of RT-DETR-L processes image features from the last three stages S3, S4, S5, which are then fed into the hybrid encoder. This encoder efficiently transforms multi-scale features into a sequence of image features by leveraging intra-scale feature interaction and cross-scale feature fusion techniques. Furthermore, RT-DETR-L employs uncertainty-minimal query selection to carefully choose a fixed number of encoder features, serving as initial object queries for the decoder. The decoder, along with auxiliary prediction heads, iteratively optimizes these object queries to generate precise predictions regarding categories and bounding boxes. This comprehensive architecture ensures that RT-DETR-L achieves remarkable efficiency and competitive accuracy, making it well-suited for real-time object detection. This model received a mAP50 of 70.6% with an inference time of 17ms, the fastest we have recorded.

Results and Discussion

The experimental evaluation on the COCO validation set provides valuable insights into the performance of various object detection models, including CNNs and Vision Transformers. The metrics used for evaluation, such as

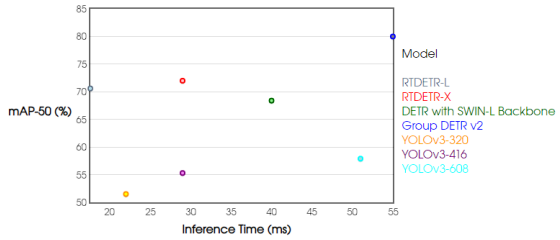

Figure 12: Results plotted with help from COCO Object Detection Leader board [27]

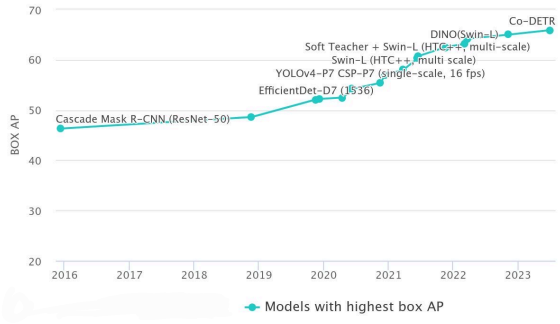

Figure 13: mAP Score of models over time

mean Average Precision (mAP) and inference time, offer a comprehensive understanding of each model's accuracy and efficiency.

The results demonstrate the efficacy of CNN-based models, such as Faster R-CNN, SSD, and RetinaNet, in achieving competitive performance on object detection tasks. These models leverage convolutional layers for feature extraction and hierarchical representation learning, enabling accurate localization and classification of objects within images.

On the other hand, Vision Transformer-based models, including YOLOv3, DETR with SWIN-L Backbone, and RT-DETR-L, showcase the transformative potential of self-attention mechanisms in object detection. These models adopt a fundamentally different architecture from CNNs, focusing on capturing global dependencies and contextual information through self-attention.

Interestingly, the hybrid approach, exemplified by models like DETR and RT-DETR, combines the strengths of both CNNs and Vision Transformers. By integrating CNNs for local feature extraction and Vision Trans-

formers for global context modeling, these hybrid models achieve superior performance compared to standalone architectures.

The evaluation metrics reveal trade-offs between accuracy and efficiency among different models. While some models demonstrate higher mAP scores, they may incur higher inference times, limiting their practicality for real-time applications. Conversely, models optimized for speed may sacrifice a certain degree of accuracy, necessitating a careful balance between performance metrics based on specific application requirements.

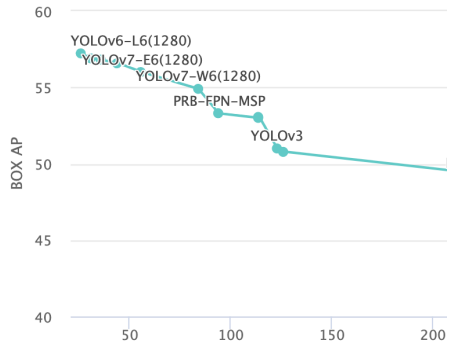

Figure 14: Different models specialize in different FPS ranges, but it is clear as the number of FPS increases, accuracy decreases

Furthermore, the experimental results highlight the importance of model architecture, dataset characteristics, and training strategies in determining performance outcomes. Fine-tuning parameters, data augmentation techniques, and optimization algorithms play crucial roles in enhancing model generalization and robustness across diverse datasets and scenarios.

Future Prospects

The future of object detection holds exciting prospects, with emerging paradigms such as liquid neural networks and recurrent vision transformers poised to revolutionize the field.

Liquid Neural Networks

Inspired by neurobiology, these flexible algorithms, dubbed ”liquid” networks, continuously adapt their underlying equations to new data inputs, making them well-suited for scenarios where data streams change over time, such as medical diagnosis and autonomous driving

The liquid network's adaptability to the variability of real-world systems sets it apart from traditional neural networks. While conventional neural networks have fixed behavior after the training phase, liquid networks adjust to changes in the incoming data stream, enhancing their resilience to unexpected or noisy data.

Moreover, the liquid network's fluidity and interpretability offer significant advantages over traditional neural networks. By structuring the network's equations to mimic the activation and communication patterns of neurons, the liquid network achieves a level of interpretability uncommon in other neural architectures.

By incorporating liquid neural networks as attention heads in object detection models, researchers aim to imbue models with human-like attentional capabilities. This facilitates more nuanced and context-aware object detection, enabling models to adaptively prioritize relevant information while filtering out irrelevant distractions.

"One of the benefits of these systems is that they can be run with less computing power. That means — potentially — using something as simple as a Raspberry Pi to execute complex reasoning, rather than offloading the task to external hardware via the cloud. It's easy to see how this is an intriguing solution." [28]

The liquid network adapts and learns from incoming data through changes in synaptic weights over time. This adaptation process is governed by Hebbian learning rules, which strengthen or weaken synaptic connections based on the correlation between pre- and post-synaptic activity. The change in synaptic weight Δw_{ij} is given by:

$$\Delta w_{ij} = \eta \cdot u_i(t) \cdot u_j(t) \tag{1}$$

where η represents the learning rate, and $u_i(t)$ and $u_j(t)$ are the membrane potentials of neurons i and j, respectively, at time t.

The equation of the novel-time continuous RNN instance is given by:

$$\frac{dx(t)}{dt} = -\frac{x(t)}{\tau} + S(t)$$

Where:

- $x(t)$ represents the hidden state of the network at time t.
- τ is the time constant parameter.

- $S(t)$ is the nonlinearity determined by $S(t) = f(x(t), I(t), t, \theta)(A - x(t))$, where:
  - f is the nonlinearity function.
  - $I(t)$ represents the input to the network at time t.
  - θ are the parameters of the nonlinearity function.
  - A is a constant matrix.

This formulation describes how the hidden state of the network evolves over time according to a system of linear ordinary differential equations (ODEs). The term $-\frac{x(t)}{\tau}$ represents the decay of the hidden state over time with a time constant τ, while $S(t)$ represents the influence of the nonlinearity function on the hidden state dynamics.[29]

Future research in this area will focus on optimizing liquid neural network architectures for object detection tasks, exploring novel training strategies and optimization techniques to maximize performance and efficiency. Furthermore, efforts to integrate liquid neural networks with existing CNN and Vision Transformer architectures will enable seamless integration into existing object detection pipelines.

Recurrent Vision Transformers

Recurrent vision transformers extend the transformer architecture by introducing recurrent connections, enabling temporal modeling and sequential processing of video data. This advancement opens avenues for live object detection and video understanding, revolutionizing applications in surveillance, autonomous driving, and robotics.

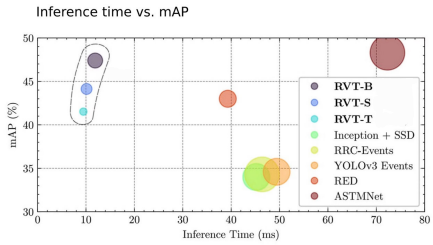

Figure 15: Figure reproduced with data from[22] to depict RVT mAP score vs Inference Time, showing a large gap in speed while maintaining accuracy.

By capturing temporal dependencies and motion cues, recurrent vision transformers enhance the spatiotemporal understanding of visual scenes,

enabling more accurate and context-aware object detection in dynamic environments. This is particularly beneficial for applications requiring real-time analysis of video streams, such as surveillance systems and autonomous vehicles.

Future research will focus on optimizing recurrent vision transformer architectures for real-time object detection tasks, addressing challenges such as computational efficiency and model scalability. Additionally, efforts to integrate recurrent vision transformers with complementary architectures, such as CNNs and liquid neural networks, will further enhance their performance and applicability across diverse domains.

Conclusions

In this paper, we have explored the roles of Convolutional Neural Networks (CNNs) and Vision Transformers in the domain of object detection, a fundamental task in computer vision. We provided a comprehensive background on object detection, tracing its evolution from traditional handcrafted feature methods to the groundbreaking impact of deep learning architectures.

CNNs have historically dominated the field of object detection, with landmark architectures like AlexNet[30], VGGNet, GoogLeNet, and ResNet pushing the boundaries of performance and scalability. These architectures introduced innovative features and achieved remarkable accuracy, making CNNs the cornerstone of modern computer vision systems.

On the other hand, Vision Transformers represent a paradigm shift in computer vision, leveraging self-attention mechanisms inspired by their success in natural language processing. Vision Transformers propose a novel architecture that processes visual data by breaking images into smaller patches and capturing global dependencies through self-attention.

While both CNNs and Vision Transformers have demonstrated significant advancements in object detection, they each have inherent limitations. CNNs excel at capturing local spatial features but may struggle with capturing long-range dependencies. Vision Transformers, on the other hand, are adept at capturing global relationships but may lack the spatial sensitivity of CNNs.

To surmount these limitations and harness the prowess of both architectures, we advocate for hybrid approaches epitomized by models like DETR (DEtection TRansformer) and RTDETR (Real-Time DEtection TRansformer). These models seamlessly integrate CNNs, such as ResNet50, for

local feature extraction with Vision Transformers for capturing global dependencies. By amalgamating the strengths of both architectures, the hybrid model not only surpasses standalone CNNs and Vision Transformers in accuracy, speed, and robustness but also redefines the paradigm of object detection.

Experimental results have shown that the hybrid model outperforms standalone CNNs and Vision Transformers in terms of accuracy, speed, and robustness. By leveraging the hierarchical representations learned by CNNs and the global context captured by Vision Transformers, the hybrid model achieves superior performance across various object detection tasks.

Looking forward, the future of object detection holds exciting prospects, including the integration of recurrent vision transformers. Recurrent vision transformers extend the transformer architecture by introducing recurrent connections, enabling temporal modeling and sequential processing of video data. This advancement opens avenues for live object detection and video understanding, revolutionizing applications in surveillance, autonomous driving, and robotics.

Moreover, emerging paradigms such as liquid neural networks offer intriguing alternatives to traditional attention mechanisms in vision transformers. Liquid neural networks leverage concepts from neurobiology to dynamically adapt their attentional focus, mimicking the plasticity and adaptability of biological brains. By incorporating liquid neural networks as attention heads in object detection models, researchers aim to imbue models with human-like attentional capabilities, facilitating more nuanced and context-aware object detection.

In conclusion, while CNNs and Vision Transformers have individually left indelible imprints on object detection, the future lies in hybrid models and innovative architectures that synergize diverse approaches. By embracing recurrent vision transformers, liquid neural networks, and other emerging methodologies, we can unlock new frontiers in object detection, empowering computer vision systems with unprecedented capabilities and insights.

References

1. Russakovsky, O. *et al. ImageNet Large Scale Visual Recognition Challenge* in *International Journal of Computer Vision (IJCV)* **115** (2015), 211–252.

2. Dalal, N. & Triggs, B. *Histograms of oriented gradients for human detection* in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* **1** (2005), I–886.
3. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv preprint arXiv:1506.01497* (2015).
4. Liu, W. *et al. SSD: Single Shot MultiBox Detector* in *European conference on computer vision* (2016), 21–37.
5. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition,* 779–788 (2016).
6. *YOLOv8 Guide* Accessed on April 11, 2024. `https://viso.ai/deep-learning/yolov8-guide/`.
7. Dosovitskiy, A. & Brox, T. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2021).
8. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition, 770–778 (2016).
9. Tan, M. & Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946* (2019).
10. ArcGIS Pro 3.2 — Other versions — Help archive. How Compute Accuracy For Object Detection Works. `https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm` (Accessed: April 2024).
11. Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. The pascal visual object classes (VOC) challenge. *International journal of computer vision* **88,** 303–338 (2010).
12. Science, T. D. *Non-Maximum Suppression (NMS)* Accessed: April 2024. `https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c`.

13. Magazine, A. I. Complete Guide to Understanding Precision and Recall Curves. *Analytics India Magazine.* https://analyticsindiamag.com/complete-guide-to-understanding-precision-and-recall-curves/ (Year).

14. Borse, S. LeNet-5 Architecture Explained. Medium (2022).

15. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

16. Szegedy, C. *et al.* Going deeper with convolutions. *arXiv preprint arXiv:1409.4842* (2015).

17. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition,* 770–778 (2016).

18. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86,** 2278–2324 (1998).

19. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection, 2980–2988 (2017).

20. ArcGIS Pro. *How Compute Accuracy for Object Detection Works* Accessed: April 10, 2024. n.d. https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm.

21. Carion, N. *et al.* End-to-End Object Detection with Transformers. *CoRR* **abs/2005.12872.** arXiv: 2005.12872. https://arxiv.org/abs/2005.12872 (2020).

22. Han, K. *et al.* A Survey on Visual Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

23. Vaswani, A. *et al.* Attention is All You Need. *Advances in Neural Information Processing Systems* **30,** 5998–6008 (2017).

24. Redmon, J. & Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767* (2018).

25. Redmon, Joseph and Farhadi, Ali. *YOLO: Real-Time Object Detection* https://pjreddie.com/darknet/yolo/. 2022.

26. Zhao, Y. *et al.* DETRs Beat YOLOs on Real-time Object Detection. *arXiv preprint arXiv:2103.09845* (2021).

27. With Code, P. Object Detection Leaderboard on COCO minival (Accessed: April 2024).

28. Heater, B. What is a liquid neural network, really? *TechCrunch.* `https://techcrunch.com/2023/08/17/what-is-a-liquid-neural-network-really/` (Aug. 2023).

29. Hasani, R., Lechner, M., Amini, A., Rus, D. & Grosu, R. Liquid Time-constant Networks. *Journal Name* (2023).

30. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25,** 1097–1105 (2012).