Shah and Tembhurne
Journal of Electrical Systems and Inf Technol (2023) 10:54
https://doi.org/10.1186/s43067-023-00123-z

Journal of Electrical Systems and Information Technology

REVIEW Open Access

Object detection using convolutional neural networks and transformer-based models: a review

Shrishti Shah[1] and Jitendra Tembhurne[1*]

*Correspondence: jtembhurne@iiitn.ac.in

[1] Indian Institute of Information Technology, Nagpur, Maharashtra, India

Abstract

Transformer models are evolving rapidly in standard natural language processing tasks; however, their application is drastically proliferating in computer vision (CV) as well. Transformers are either replacing convolution networks or being used in conjunction with them. This paper aims to differentiate the design of convolutional neural networks (CNNs) built models and models based on transformer, particularly in the domain of object detection. CNNs are designed to capture local spatial patterns through convolutional layers, which is well suited for tasks that involve understanding visual hierarchies and features. However, transformers bring a new paradigm to CV by leveraging self-attention mechanisms, which allows to capture both local and global context in images. Here, we target the various aspects such as basic level of understanding, comparative study, application of attention model, and highlighting tremendous growth along with delivering efficiency are presented effectively for object detection task. The main emphasis of this work is to offer basic understanding of architectures for object detection task and motivates to adopt the same in computer vision tasks. In addition, this paper highlights the evolution of transformer-based models in object detection and their growing importance in the field of computer vision, we also identified the open research direction in the same field.

Keywords: Convolutional neural network, Object detection, Transformer-based attention, Faster R-CNN, Semantic segmentation, Segmenter, YOLO's

Introduction

Object detection (OD) is growing rapidly due to the rebirth of convolution neural networks. The deep CNNs are capable to learn prominent-feature representations of images due to their typical hierarchical architecture, and hence, it offers a fast, rapid, and accurate way to predict the position of objects within the image. Moreover, Recurrent-CNN (R-CNN) [1] accomplished the noteworthy success in CV tasks, as CNN categorizes the class of object only but not capable in determining the position or location of object in the given image.

Due to the few number of unsolved challenges and slow computing nature of OD, R-CNN is revised, and we witnessed the changes in R-CNN for object detection models such as fast R-CNN [2], faster R-CNN [3], region-based fully convolutional networks

(R-FCN) [4], single-shot detector (SSD) [5], you only look once (YOLO) [6], spatial pyramid pooling network (SPP-Net) [7], and mask R-CNN [8]. These new models are proved successful in computing better in terms of results and accuracy. It is noticed that object detectors are categories as—(1) one-stage detector, such as SSD and YOLO, which achieves high inference speed, and (2) two-stage detector (such as R-CNN, fast R-CNN, faster R-CNN) that provide high localization and better object recognition accuracy. One-stage detectors compute the prediction boxes directly from the given input images, without applying region proposal stage, and thus, it becomes time efficient which can be employed for real-time problems. On the other hand, the two-stage detector provides bounding boxes by applying region proposal network (RPN), which is followed by feature extraction stage.

In general, these new models ignite a need for their application for real-world problems, and led to well-researched domains in the field of image object detection, which includes pose detection, face detection [9], people detection [10], crowd detection [11], traffic sign detection [12], pedestrian detection [13], etc. For the purpose of applying these models in a particular field, there should be a pure understanding of particular model so that it becomes easy to adopt, along with technique of its application. In addition to this, the models also led to new models of object detection and image classification. As a consequence, it helps in improving the applications of multi-region detection [14], instance segmentation [15], edge detection [16], salient object detection, action recognition [17], fault detection, text recognition, etc.

The new approaches and models are continuously evolving; therefore, in this paper, we aim to show the comprehensive study on transformer-based detectors that establishes a powerful backbone for visual recognition, and proved to achieve competitive results in contrast with convolutional networks. Here, the new architectures along with old paradigm are explored, and similar architectures using different modules that achieve better competitive results are identified. The CV community adapted transformers extensively in this field. The transformers and their variants are to be successful in CV tasks. The transformer-based models are differentiated into two categories such as single-head self-attention wherein local or global self-attention is adopted within convolutional networks. Other is multi-head self-attention that cascades multiple transformer layers.

Transformers-based modules are utilized for OD as per following sequence: (i) feature extraction by transformer backbones, along with R-CNN head for OD [18], pyramid vision transformer (ViT) [19], Twins [20], CoaT [21], Swin transformer [22], convolutional vision transformer (CViT) [23], shuffle transformer [24], CrossFormer [25], RegionViT [26], and focal transformer models [27], (ii) visual features extraction by CNN backbone and a transformer-based decoder for OD, i.e. detection transformer (DETR) [28], deformable DETR, [29], and (iii) end-to-end OD by transformer (i.e. YOLOS) [30].

During the development of a model, a new module is developed to improve and overcome difficulties and challenges in the domain. In CV, researchers designed improved CNN models to overcome the various complications encountered with the existing models for the real-world applications. The applications are face detection [31], human–object detection [32], traffic signal [33], pedestrian detection [34], etc. With exception to this, new models are also designed and integrated with base detection transformer frameworks such as vision and detection transformer (ViDT) [35], multiple object detection [36], instance segmentation [37], one-region multiple objects [38], etc. Moreover, it is observed that transformer models show a lot of potential to become a new generic detection pipeline [39].

In this work, the review of existing detectors in convolutional networks and transformer-based architectures is presented, which share the same paradigm but have different architectures. This study also enables to understand the adoption of different models for specific application and enhances the scope for further development. The motivation for this work is to offer OD understanding and applicability in various domains for researchers started investigating on transformers in vision, object detection tasks, and applications. It is noticed that a number of transformer-based models are designed for CV task, though there is a scope still exists to deploy more accurate applications having a highly precise real-time system.

Contributions

The contributions targeted in this article are as follows;

1. To investigate object detection using CNNs and transformer-based architectures, and sharing same paradigm containing diverse architectures.
2. To identify applicability and suitability of different models in object detection for specific application and further improvements for advance development.
3. To propose the insight and detailed analysis of utilization of transformers in computer vision, object detection and other similar tasks. Further, examined different transformer-based models handling CV task to offer better accuracy in highly precise real-time system.
4. To identify the issues and propose the future directions in object detection task.

The paper organization is as follows: Working of different object detection models based on RCNN, ViT, Faster R-CNN, Mask R-CNN, DETR, and YOLOs is presented in "Object detection models" section. "Tasks and model evolution" section reveals on task and model evolution. "Applications of transformers" section highlights the application of transformer in different domains. Further, "Datasets and evaluation metrics" section, proposed the datasets and various evaluation metrics for OD. "Performance analysis and discussion" section presents the performance analysis on OD. "Challenges in object detection" section summarizes the challenges in OD. Finally, conclusion and future scope are discussed in "Conclusion and future scope" section.

Object detection models

It is observed that for the task of OD, number of datasets—PASCAL VOC datasets [40], Microsoft COCO datasets [41], ImageNet datasets, [42], etc., are utilized for model training and evaluation. At present, variety of datasets being utilized for different tasks. Figure 1 shows the emergence of various datasets adopted for OD. In addition, Fig. 2 highlights the improvement in accuracy of various OD algorithms on MS-COCO, VOC07, and VOC12 datasets, since 2005 to 2021.

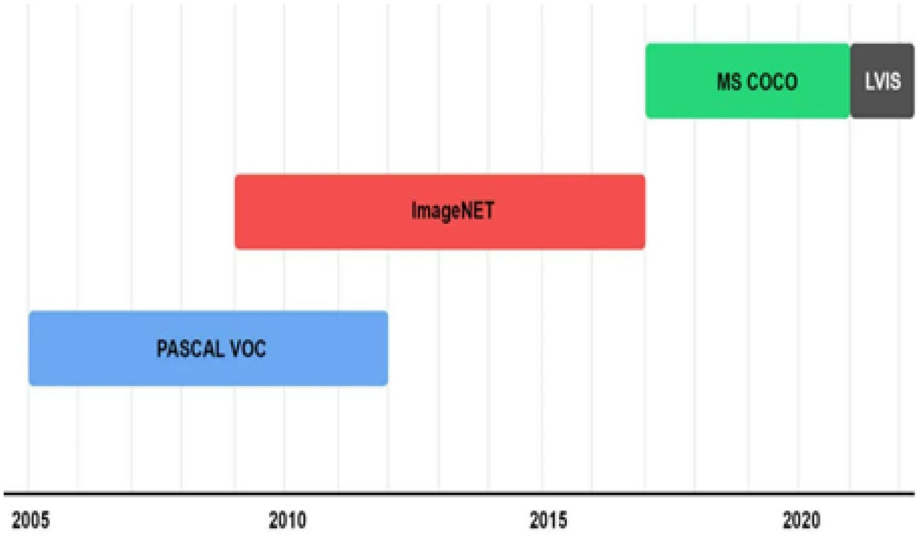


Fig. 1 Timeline of important datasets

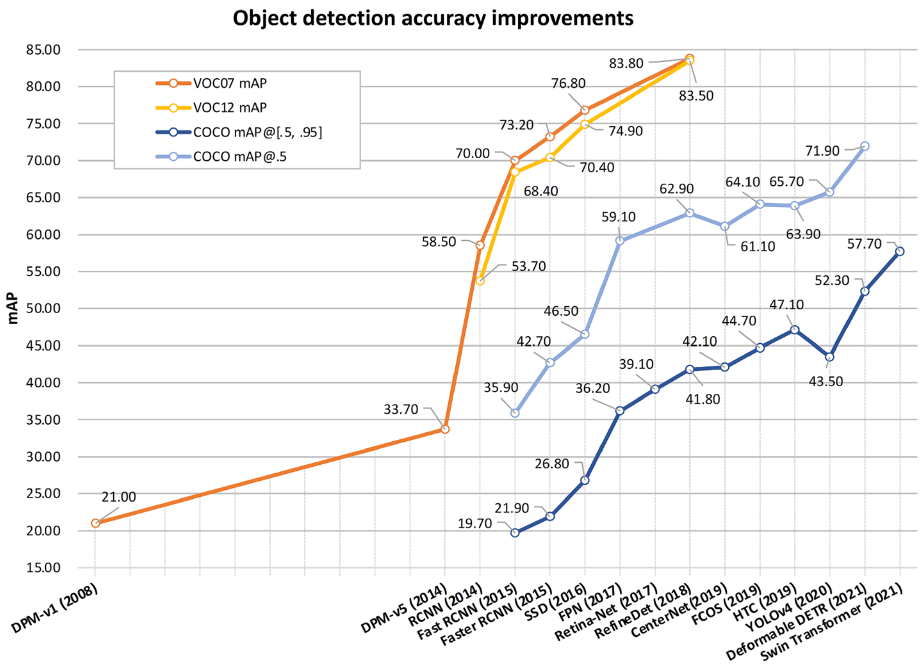


Fig. 2 OD on MS-COCO, VOC07, and VOC12 datasets [43]

The OD challenges includes—(1) non-accessibility of labelled dataset, (2) multi-scale images with different objects, (3) overlapping objects in videos, (4) low-scale video processing, (5) inherent variant in objects occupying in terms of pixel, i.e. 60–70%, 10–20%, and few pixels or less.

R-CNN

It is generally acknowledged that the progress is slowed down during the year 2010–2012, as shown in Fig. 1, with small updates in SIFT [44] and HOG [45] models. In

[1], a simple and scalable detection algorithm is proposed, which achieved mean average precision (mAP) of 58.3% (i.e. 23% higher mAP than the existing detectors). Here, high-capacity convolutional network is applied to the bottom-up region proposal. The idea is really basic, i.e. it acquired the input image, extracted around 2000 bottom-up region proposals (based on selective search), features are computed for every proposal using a large convolutional network, and then linear support vector machine (SVM) as a classifier is adopted to predict object in each region. Moreover, for recognizing object categories and after scoring each selective search, class-specific bounding box namely regressor is created for OD. Figure 3 shows the improvements in mAP on PASCAL VOC dataset since 2006–2016.

The RCNN model consists of three modules, which is presented in Fig. 4:

(i) R*egion proposal* R-CNN is not a big fan of particular region proposal method, and thus, category-independent region proposals like selective search are utilized.

(ii) *Feature extractor* The extracted feature vector from the region proposal is transformed into 227 × 227 RGB colour plane due to its compatibility with CNNs. Then, forward propagation is achieved through five convolutional layers and two dense layers for feature calculation.

(iii) *Test time detection* Each class is scored upon extracted features using SVM-trained classes. By giving each score, bounding boxes are derived through greedy non-maximum suppression.

The RCNN model achieved boost in performance and a large improvement of mAP through two comprehensions—(1) applying high-capacity CNNs to bottom-up region proposals to localize objects, and (2) train large CNNs when training data is labelled and uncommon thus pre-training the model for image classification, later fine-tuning the model for detection task. Although RCNN made a great progress, feature extractions on large amount of overlapped proposals (i.e. more than 2000 boxes per image) lead to very slow detection (14 s. per image on the GPU). As RCNNs are two-staged object detectors, the highest obtained detection accuracy is still slower.

Further, it is observed that fast R-CNN is developed with the intention to reduce the training time, as it runs the CNN once on the whole image instead of computing each 2000 region of interest (RoI) from region proposal, individually. The end layer of deep

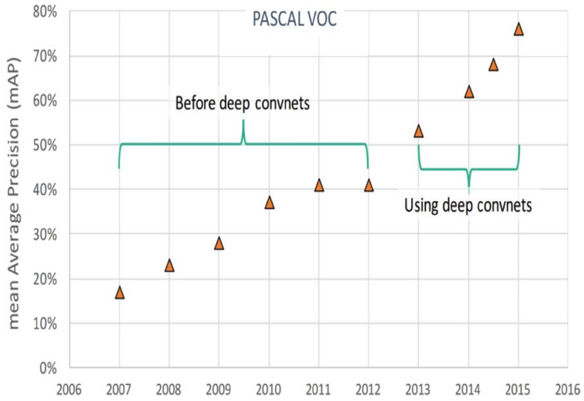
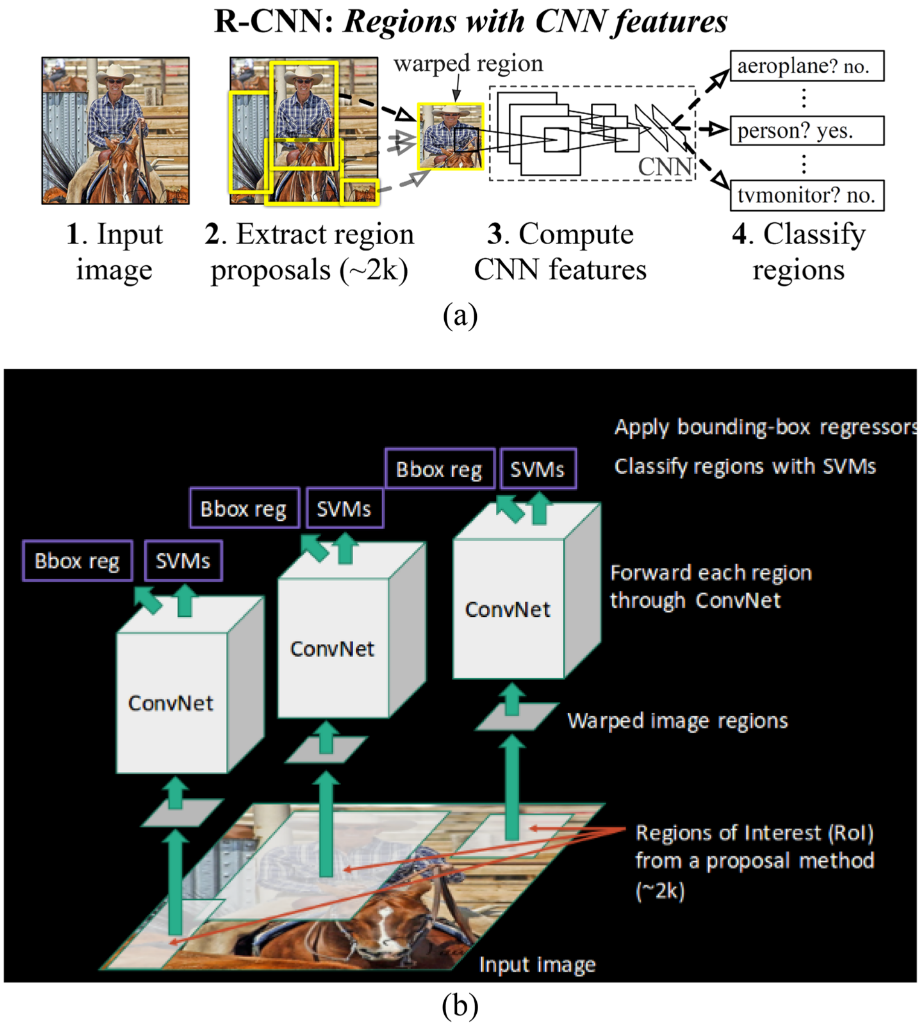


Fig. 3 Timeline of mAP on PASCAL VOC dataset


R-CNN: *Regions with CNN features*
warped region
aeroplane? no.
person? yes.
tvmonitor? no.
CNN
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions
Apply bounding-box regressors
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Forward each region through ConvNet
ConvNet
ConvNet
ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

(a)

(b)

Fig. 4 The detailed architecture of RCNN [2]. **a** RCNN step-by-step process. **b** Feature extraction from RoI

CNN is called as RoI pooling layer, which extracts specific features from input region. The CNN output is now realized by dense layer. Now, model produces two outputs—(1) class prediction via softmax function and (2) linear output pertaining to bounding box. This process is repeated recursively for each RoI for a given image. It is successful in overcoming the drawback of RCNN because it increases the mAP from 58.3 to 70%.

RCNN marked a remarkable success, as it is among the first to detect position or location of object, and work suitably for multiple objects images. OD in real time is difficult to accomplish because it takes testing time of 47 s, approximately for every image, thus, training system pipeline is tough. Due to these limiting factors, further improvement is still needed in RCNN.

Faster R-CNN

Faster R-CNN was introduced as a first near real-time and end-to-end deep learning (DL) detector for object detection. Before the introduction of this network, the region-based CNN models are computationally very costly, where the basic difference is the model utilized region proposal method to create sets of regions. The test time, which is accelerated to near real-time, exposing region proposal computation as a bottleneck with low computational time, and achieved the mAP of 69.9%. Figure 5 shows the model of faster R-CNN, comprising two modules:

i) Region Proposal Network of CNN: It is present to propose the regions and then predicts the object bounds, object scores for every position. This acts like the attention model, which is discussed earlier to inform the network where to pay more attention. Faster R-CNN network is developed for extracting features, and then working on the region proposal and producing—(1) class labels and (2) bounding box. The RPN is applied to generate region proposals; here, the model slides a small network upon feature map generated by convolutional network. This small network accepts spatial window ($n \times n$) as an input corresponding to feature map generated by convolutional network. Every sliding-window mapping is performed with lower-dimensional feature due to mini-network functions in a sliding-window manner, and dense layers sharing is confirmed among all spatial locations. To train RPN, labels assigned under binary class, i.e. object or not object to each of the anchors (here negative mining is simply balancing by weights), and selecting anchor such that highest intersection upon union overlapping corresponding to ground truth box (for this purpose, greedy selection non-max suppression (NMS) method is utilized). This feature further supplied to two sibling dense layers, i.e. box classification and box regression, also known as REG and CLS. Thus, we conclude that this model enables a unified, DL object detection system, which runs at approximately in real-time mode. RPN further improved the quality indirectly which in turn enhancement in detection accuracy is witnessed.

After faster R-CNN's bounding box regression wherein through initial proposal, predicted bounding box's location is refined, or anchor box will not assist for post-processing block. However, it is integrated with detector and training is performed in end-to-end manner to achieve better prediction and smooth functioning.

ii) Sharing Features for RPN and Fast R-CNN: This network layer does not consider the region-based OD CNN instead adopt fast R-CNN.

The training of fast R-CNN and RPN is accomplished independently, the modification in convolutional layers is achieved differently. Therefore, technique needs to be developed for allowing sharing of convolutional layers amid two networks. The second-stage detector makes predictions relative to some initial guesses made by the RPN, whereas

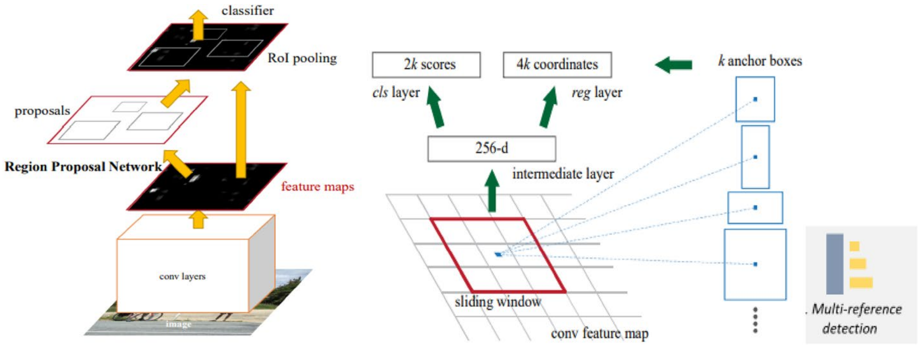


Fig. 5 Detailed architecture of faster R-CNN [3]

single-stage methods use anchor boxes. Nevertheless, the recent one demonstrates that its final performance heavily depends on initial guesses. This is done by directly setting the losses wherein no post-processing is required. Thereafter, the handcrafted procedure is removed and detection is streamlined through the direct prediction via absolute box prediction for the set of detections.

Vision transformer (ViT)

In [46], a new research direction is offered wherein the model is developed as a competitor to the CNNs. It also attained an excellent results while requiring substantially a fewer computational resources for the training purpose (i.e. almost four times in terms of computational efficiency and accuracy). Subsequently, it is witnessed that the transformer models made remarkable existence in natural language processing (NLP), as they are solely based on attention mechanism. It is seen in transformer model [47] that it is a combination of various attention mechanisms. It consists of following parts, which is also depicted in Fig. 6.

i) Attention: This is discovered to let the decoder utilize the most relevant part of the input sequence by a weighted combination of all encoded input vectors. The attention also helps the model not to forget the input and the decoder to know where to focus. Herein, each vector query, i.e. $q = s_t - 1$, previous decoders output against a database of keys to compute a score value, which is computed as a dot vector of specific query with key: $e(q,k) = q.k_i$ (captures each feature to see its relation with other features). The above score is then passed through the softmax$\alpha(q,k_i) = \text{softmax}(e(q,k_i))$, and attention is calculated by a weighted sum of vector value $v(k_i)$ in order to retain the focus on these words that are relevant to the query.

$$\text{Attention}(q,K,V) = \sum \alpha(q,k_i).v(k_i)$$

here, the attention function is described as—(1) a query and (2) set of (key, value) pairs corresponding to output, and all the parameters are vectors. The output computation comprises weighted sum wherein weight assigned (value) is calculated using compatibility function related to query along with the selective key. In a pictorial representation, if a single object is considered, the attention between patches containing parts of the same object will be high rather than a patch containing background and another containing an object.

ii) Self-Attention: It is proposed due to the challenges faced by the encoder–decoders in dealing with long sequence models. In the attention model mechanism, the output of

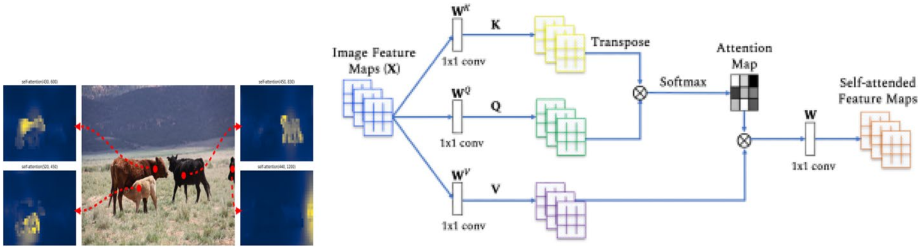


Fig. 6 The detailed architecture of vision transformer [48]

the decoder focuses attention mainly on the input, whereas in self-attention models, the input to interact with each other is now allowed.

The scaled dot product attention initially computes a dot product for each q and k, it divides each result by $\sqrt{dk}$, so that after the dot product cause some self-attention scores to become small. Hence,

$$\text{Attention}(Q, K, V) = \sum \left(\frac{\alpha(Q, Kt)}{\sqrt{dk}} \right) v(k_i).$$

In transformer models, relevance of one item with other items is computed using self-attention. Further, self-attention layer updating each computation in sequence by applying global aggregating on finished input sequence, and hence, it does not forget and capture the interaction among all the entities of input.

(iii) Self-Attention in Vision: This model allows long-term dependencies while handling the sequence elements, and proves to be better than CNNs (which needs large receptive fields). A single head of self-attention works similar to that of the above model, it gives input sequences of image features, i.e. all pixels in the given patch, computes q, K, and V vectors, and aggregated spatial information which is identified within the patch. The V vectors aggregation is performed after projecting softmax score of q and K, and triplet (key (K), query (q), value (V)) is calculated, which is followed by attention computation. Thereafter, applying it to reweight the values, the output projected is employed to find output features confirming same dimension as that of input.

The feature maps input to self-attention, compute their response at a position (i.e. positional embedding), and make it possible to capture relations or connections between any two locations or positions in the map. This is irrespective of distance; hence, information is integrated across the image for lowest layers as well.

Self-attention has two types to implement vector attention which learns weight corresponding to channel and spatial dimensions—(1) pairwise and (2) patch-wise. The pairwise self-attention computes vector attention as a relationships of feature corresponding to neighbours in a particular local neighbourhood. On the other hand, self-attention using patch-wise mechanism offers generalization using convolution operator. Eventually, the model applied over image regions is found, which are semantically significant for classification. Therefore, as a concluding part, it is computationally intensive and allows to capture long-term interaction, also and focus on the importance of particular feature. Explicitly modelling all the pairwise relations between elements in the sequence, thus, makes it suitable for specific constraints such as removing duplicates. Moreover, self-attention is utilized as one of the layers in the object detection transformer model. Attention is applied in some layers and used to connect two modalities like the encoder to the decoder, while self-attention is applied within a component.

iv) Multi-headed attention: Fig. 7 shows the flow diagram of multi-headed attention wherein combination of n single-head self-attention, each consists of three parameter matrices of their own (i.e. weight matrices $\{Q_i, K_i, V_i\}$). Concatenation of output in context vectors of each head is the output of multi-headed attention layer. It is seen that the different learned representations can improve the transformer model. Self-attention is invariant to permutations and modification in the input points occurs when combined for multi-headed attention. It easily operates on irregular input data, unlike conventional

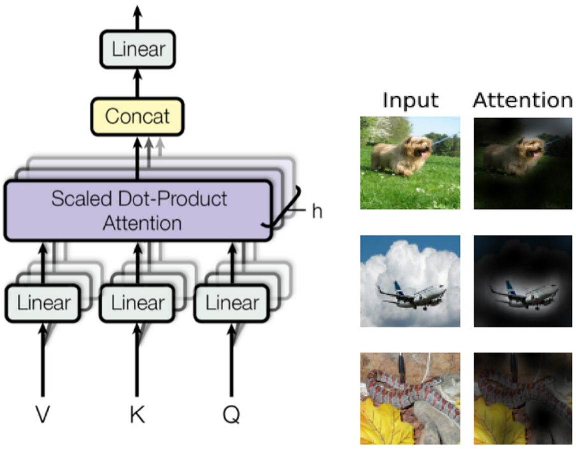


Fig. 7 Multi-headed attention scheme [46]

convolution, and requires a specific grid structure. Self-attention offers the capability to learn the features from global and local region, and hence, their experimental results confirm that the multi-headed self-attention (along with sufficient parameters) is a more generalized procedure.

The transformer model consists of encoder–decoder structure wherein inputs and outputs sequences are supplied with one element at a time.

v) Encoder: The encoder is a stack of identical layers ($N=6$) wherein every layer consists of two sub-layers. First layer is a multi-headed self-attention, each of these layers contains all the queries, keys, and values that belongs to same place. Here, previous layer output belongs to encoder, and every position in encoder attends to all the positions in previous layer in encoder. Further, second layer is a position-wise dense layer feed-forward network.

vi) Decoder: The decoder is also a stack of identical layers ($N=6$). Decoder consists of three sub-layer, two sub-layers are similar to encoder wherein third layer performs multi-headed attention on output of encoder stack. Similar to encoder, residual connection is employed around each of sub-layers (i.e. self-attention layers) which is followed by the layer normalization.

This transformer model replaced the CNNs backbones in OD models, some models improved their detection accuracy by combining various feature maps [48] in multi-headed self-attention scheme. The target detection performance is drastically improved by integrating feature maps output from CNNs, or already existing object detectors with multi-headed attention, or attention modules fusion features. Sometimes, the encoder module is replaced or integrated, while at some places, the decoder module is replaced; however, the structures and major idea behind functioning remain the same. Minimal considerations that are adequate to overcome the aforesaid challenges are required.

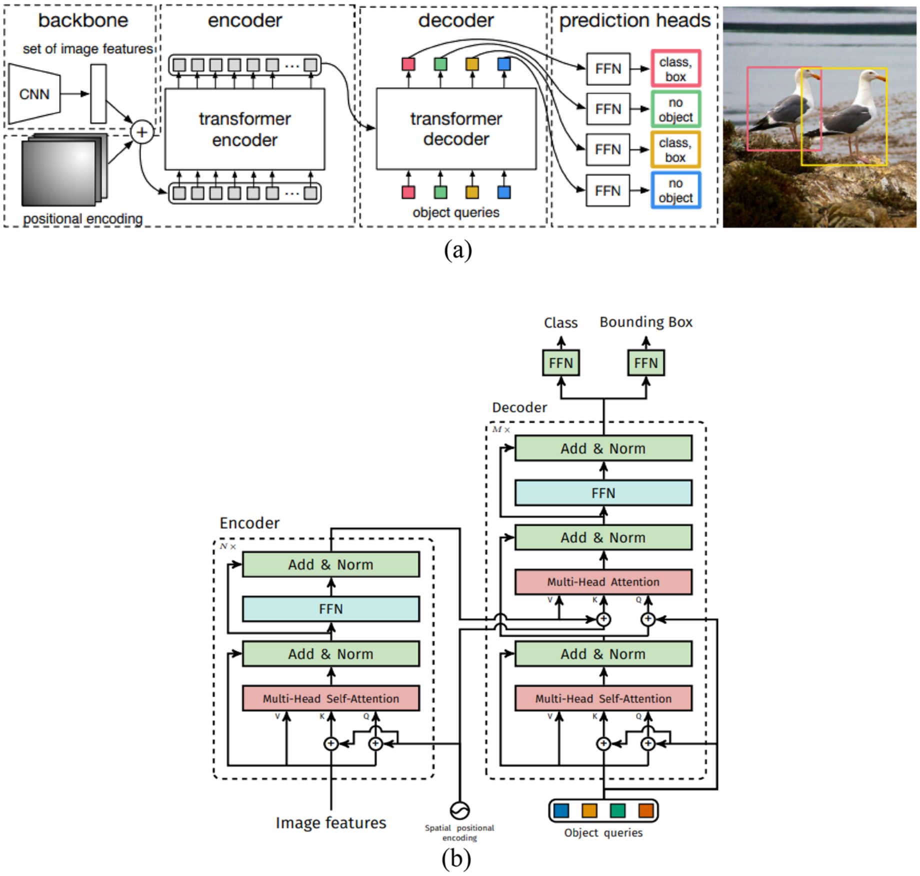
In [49], it is stated that for years, OD models rely on recognizing the object instances independently without exploring their relation during the learning of the model. Here, object relation component is proposed wherein object set is processed simultaneously through collaboration between their feature appearance and geometry, and thus, permitting modelling of their relations, improving object recognition and even removal of duplicates is also performed. This perfectly suits like the working of transformers, where demonstration explained that pure transformer employed to the sequence of image patches performs well on OD tasks. The image is extracted into patches by looping over annotations and image (linear projection of flattened patches), and positional embedding (convolution is a translation and scale equivariant while pooling is a translation and scale invariant). Both equi-variance and invariance are important for recognition, object detection, and segmentation purposes. Vision transformer is observed to become invariant to the position of patches, and hence, positional embeddings are added making it the only inductive bias, passed through the ViT model instead of CNN (like in RCNN), multi-headed attention layer is utilized for self-attention, and applied to a sequence of image patches. The encoded patches and self-attention layers outputs are normalized and fed into a multilayer perception (MLP) [50]. The MLP is used for classification head along with a hidden layer at the time of pre-training, and fine-tuning is performed by single linear layer. The MLP mixer comprises classifier head, mixer layers, and per-patch linear embeddings. In addition, channel-mixing and token-mixing MLP are the part of mixer layer wherein each consisting of two dense layers and GELU. The unused outputs correspond to input patches from MLP layer (ViT classifier) that can encode local information, which is beneficial for performing OD. The model outputs the four dimensions representing the bounding box coordinates of an object.

Subsequently, it is found that CNN lacks in global understanding of the image. To track down long-range dependencies within the image, CNN needs large receptive fields, whereas ViT model for object detection performs better than CNN [51]. The model advantage over RCNN is that it is pre-trained on large textual corpus, further, fine-tuned on dataset of smaller task, which leads to computational efficiency and scalability. It also supports multi-scalable features, due to densely vision tasks generally involve visual object's understanding with different size and scale. Transformers can be pre-trained on data of enormous amount, further, applied to specific smaller tasks through fine-tuning. The high computational complexity of self-attention and attention models indicates that there is a limitation of low-resolution inputs. Hence, few applications of CV might contain some limitations with transformer models [52].

Detection transformer (DETR)

The architecture of DETR shown in Fig. 8 is comparatively simpler than all the previous transformer-based architectures, which contains all kinds of engineering hurdles, thresholds, hyperparameters, and are unable to become competitive with stronger baselines. The design of DETR proposed a direct set prediction problem with unique predictions via bipartite matching, which uses transformers encoder–decoder model.

DETR directly predicts set of detections by combining CNN and transformer model. CNN utilized to learn 2D representation, flattening and positional encoding is performed before the transformer encoder. The transformer decoder decodes the input, i.e. learned positional embeddings, namely, object queries. Further, decoder output (i.e. embedding) is supplied to feed-forward network (FFN) to detect "object" (class and bounding box) class or "no object" class. Using a self-attention encoder–decoder model upon these embeddings makes the decision that objects together persisting pairwise



(a)

(b)

Fig. 8 The architectural details of DETR [28]

relation. Parallelly, whole image is considered as context and also makes these models suitable for set prediction under constrained environment, i.e. removal of duplicate predictions.

The main components are described as follows:

(i) *CNN Backbone*: Compact features are extracted (a lower-resolution activation map, and 2D representation of an image), where channel dimension is reduced by 1 × 1 convolution for high-level activation map to smaller dimension by constructing new feature map by $z_o \in R^{d \times H \times W}$. The model shown here is flattening and supplementing it with a position encoding before encoding using transformer encoder.

(ii) *Transformer Encoder:* It consists of six standard encoder layers wherein each layer made up of multi-headed self-attention and FFN. Like, the encoder module of ViT, its module also helps to understand the patches with each other globally (which is possible through global scene reasoning), and thus, it is important for disentangling objects. It also seems to separate cases that simplifies the extraction of object and decoder's localization. The spatial dimension of z_o (i.e. output of CNN) is also collapsed into one dimension. Here, spatial resolution of encoder plays vital role in determining OD performance, i.e. significant AP gain is achieved by spatial features resolution increment. This is comparatively better than the CNN output feature maps for sliding window. The output of encoder layer is reinterpreting the final transformer states (encoder from ViT), however, eliminating class token instead of outputting spatial feature map.

(iii) *Transformer decoder*: It consists of six standard decoder layers, each of which has embedding of size "*d*" with multi-headed, self-encoder–decoder attention at every layer. Decoding performed on positional embeddings using transformer decoder. Unlike, the attention model, it is not an autoregressive model and decodes the objects parallelly. Consequently, detection pipeline is simplified by dropping components of multiple hand-designed—encoded with prior knowledge, such as non-maximal suppression or spatial anchors. Here, the improvement, which is included by NMS, diminishes with increase in depth. Due to self-attention upon activation function allowing the model to prevent duplicate predictions. Prediction FFNs and Hungarian loss after every decoder layer are added. All prediction FFNs share parameters, which forecast the detection of an object. The transformer decoder is basically replacing the greedy selection and RoI pooling of faster R-CNN.

Some of the unique features of DETR are as follows:

1. DETR model as compared to previous models that work on direct set prediction contains an extra feature, i.e. conjunction of bipartite matching loss.
2. Unlike, other transformers, it performs non-autoregressive decoding.
3. It is equivalent to faster R-CNN training, which balances the proposals in positive/negative by subsampling. Matching cost considers both the classes' predictions and similarity in prediction and ground truth boxes.

DETR model thus attains comparable performance against the competitive faster R-CNN. The architecture is similar to few modules that is replaced by the transformer encoder–decoder. DETR exhibits expressively better results on objects in large size, the outcome likely to be achieved by non-local computations of transformer. However, the drawbacks of this model required more training epochs compared to typical detectors to converge, and achieved relatively low performance for small objects.

Deformable DETR

It consists of deformable attention unit learns to attend sampling locations within feature map, and hence capable in processing high-resolution feature maps. It uses RelationNet (RN) [53] and non-local networks (NLN) to form attention amid pixel features and bounding box features for the purpose of OD.

Toward transformer-based object detection

However, DETR models perform encoding of visual features using CNNs, whereas transformers are utilized to decode features into OD outputs. However, transformers adopted for encoding visual features while the RPN model applied for detecting outputs are surveyed successfully. It usually adds a detection network to a ViT. The ViT utilizes merely the state analogous to input class token on final layer and is outputted through MLP classifier head. It is resemblance to transformer encoder of DETR. In this model, the remaining tokens which are only utilized as features to attend the class token are applied to encode the local information. This corresponds to the input patches and are used for detection, as it outputs a visual representation of patches if image is considered globally. By reinterpreting outputs in spatial manner, feature map is created that certainly lends itself as an input to detection model similar to faster R-CNN. Moreover, detection network of faster R-CNN contains RPN that densely predicts the presence of object. The features correspond to top region proposals, further, RoI pooled and supplied to detection head, and classification is performed for each region, in addition, coordinates of bounding box are confirmed by regresses. Herein, RPN predicts the region with objects by generating several predictions per location from feature maps, which is outputted by the encoder. Predictions are employed to mark the territory in the form of anchor boxes of varying sizes and aspect ratios. Generating one feature per region proposal is the RoI pooling that is mainly achieved and at the end. These pooled features are passed through the pairs of heads, one for classification of the object detected and the other is bounding box regressor. The detection network works exactly like the transformer decoder in DETR.

Figure 9 shows the detailed flow diagram of toward transformer-based OD, which is a fully trainable end-to-end like DETR with few numbers of added adaptations and replacements. Transformer demonstrates excellent performance in pre-training of a large dataset and transferred to fewer data points via fine-tuning. This is observed before as well, however, when training is performed upon strong visual representation, it is found that network based on transformer can be developed for certain visual tasks. Further, resolution of input image limits the performance. Thus, this model is quite competitive with the DETR model.

Authors in [18] also investigated various methods of using features of intermediate encoder (input) to detection network, wherein output obtained from last layer of transformer is utilized and concatenation is performed with all state of intermediate transformer. Finally, it is found to become helpful to augment intermediate residual blocks among spatial feature map of encoder and detection module, thus, AP gain is achieved (it indicates that pre-trained transformer for classification only not adequate for detection task). Here, conclusion is also made that transformer backbone combined with different modules of CNNs models can possibly make progress on complex vision tasks because it can pre-train on large data and fine-tune on new data with improved complications. As a consequence, model is capable to handle wide array of vision-related problems. However, recent research recommends that OD can be

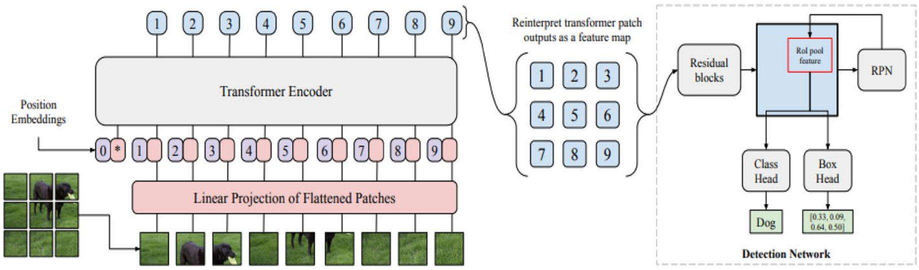


Fig. 9 Architecture of toward transformer-based object detection [18]

enhanced by adopting semantic segmentation (SS), where object boundaries are well encoded along with accurate object localization that learned with segmentations.

Mask R-CNN

It is noticed that SS detects all present objects at the image pixel level. On the other hand, instance segmentation identifies every object instance corresponding to each object of image. The instance segmentation is able to improve the visual understanding of the surround world and gained huge attention in various CV applications. Its main objective is to perform classification of each pixel into different categories.

Faster R-CNN is not intended for detecting alignment in pixel-to-pixel; hence, it is extended by adding predicting segmentation masks for every RoI (possible by utilizing small FCN, and segmentation mask is predicted based on pixel-to-pixel basis) in parallel with classification and bounding box regression. Faster R-CNN after RPN, extracting features by RoIPool from every candidate box and perform classification and bounding box regression. Faster R-CNN contains two outcomes for every candidate object, i.e. offset of bounding box offset and class label. Now, third branch that produces object mask is added to this model. There are two stages with two different neural networks, first stage is faster R-CNN's basic RPN model, which is applied to extract the RoI. It helps to find the class label and computes the bounding boxes. Second stage contains a neural network with a similar procedure to that of RPN. It extracts region from the first stage and without anchor box which uses RoIAlign for locating each importance of feature map, generates pixel-wise mask.

The model of mask R-CNN is presented in Fig. 10, we also, identified some basics foundations are as follows:

i) Here, CNN is utilized to generate feature map of image. RPN utilizes CNN to produce multiple RoI by lightweight binary classifier to differentiate the existence or nonexistence of object. The output is computed by passing NMS to anchors with high scores. This is analogous to RPN module of faster R-CNN and the transformer's decoder.

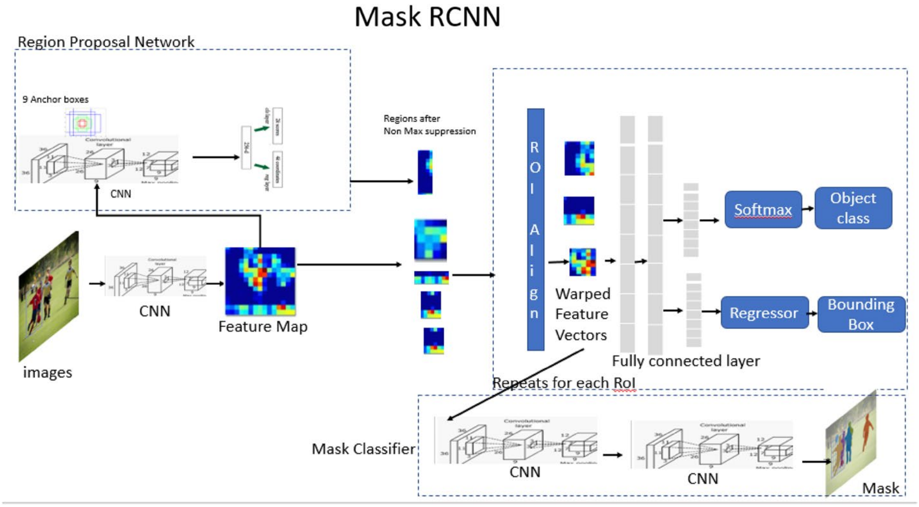


Fig. 10 Work flow diagram of Mask R-CNN [8]

Wrapped features are passed through two dense layers to make classifications and fed into mask classifiers with two FCN [54] layers for each RoI. After that every mask for each class is generated. Now, by evaluating the convolutional network on each extracted RoI makes an object detection model segmented.

ii) To make understanding of the main feature: It is observed that extracting the RoI is different here. RoIPool is employed, and then passed through RoIAlign for fast speed and better accuracy. Mask R-CNN is indirectly a faster R-CNN learning extraction of RoIs and attention mechanism using RPN. RPN-generated RoIs which are reduced to make the discrete granularity of feature maps through RoIPool perform coarse spatial quantization related to feature extraction. This quantized RoI further partitioned into spatial bins that are quantized themselves. Now, addressing misalignment issue, quantization-free layer, namely RoIAlign, tries to preserve precise spatial locations with no loss of data, and outputs multiple bounding boxes are wrapped into a fixed dimension. It is found that RoIAlign done a great impact by improving mask accuracy relatively 10–50%. Moreover, it is concluded to become vital to predict class and decouple mask. Here, binary mask for every class is individually predicted without any competition among the classes. This depends upon network's RoI classifying branch for predicting category. It is seen that stricter localization metrics for extracting features with exact spatial locations will lead to bigger gains.

Transformer models become successful in replacing the FCN models wherein stacked convolutions are employed to capture semantic information. However, self-attention is able to model the rich interactions between pixels and gives competitive outputs in comparison with CNN-based compact prediction tasks (i.e. image segmentation and semantic). Number of segmentation approaches are inserted self-attention along with CNNs. However, some recent researches proposed transformers encoder–decoder like SEgmentation TRansformer (SETR) [55], which contains ViT encoder, and two decoder designs pertaining to progressive up sampling, multi-level feature aggregation. The SegFormer [56] consists of hierarchical pyramid ViT (lacking position encoding) encoder and segmentation mask is generated by MLP-based decoder (sampling operation). Image features are extracted using segmenter (ViT encoder) and segmentation mask prediction is performed by decoder (mask transformer).

ViT segmenter

In [57], a segmenter and transformer model for the purpose of semantic segmentation is introduced. Here, semantic segmentation is utilized to make partition within image into segments to provide high-level image representations of the target task. This is accomplished by assigning each pixel of image to category label pertaining to underlying object. The approach is purely transformer-based, as it is built on ViT and extended up to semantic segmentation. The ViTs do not use CNN, however, capture contextual information through designing, and outperform the FCN-based techniques. Further, image patches are estimated to sequence of embeddings and transformer is used for encoding. The output from ViT encoder and obtained class labels from the embeddings are taken as an input by the decoder, i.e. the mask transformer. A transformer-based decoder is proposed for class masks generation, which outperforms the already existing linear baseline models, and further extended to accomplish image segmentation tasks, in general.

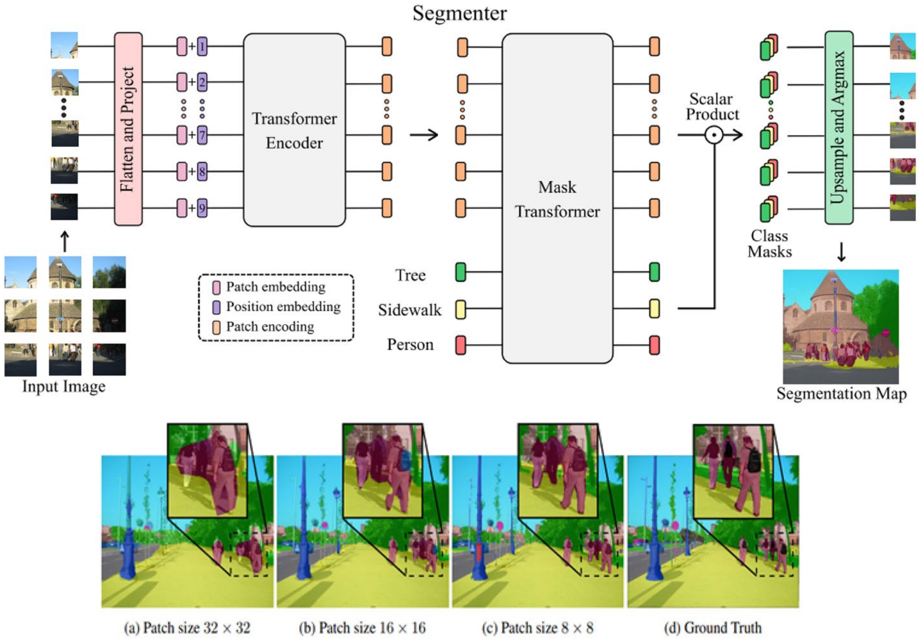
Later, model is trained in end-to-end and pixel-to-pixel with cross-entropy loss at each pixel. Therefore, the transformer decoder model can be seen to replace the FCN, but this decoder is unlike the ones who proposed their models for object classification and detection tasks.

Figure 11 illustrates the ViT segmenter model, whose encoder and decoder modules are detailed as follows:

i) Encoder: It comprises multi-headed self-attention unit wherein point-wise MLP unit (two layers with layer norm (LN)) is placed after aforementioned unit, and MLP unit employed prior to every block (like a typical encoder module). Here, each of the split image patches is flattened into 1D form, further, it is linearly projected into patch embedding along with positional embeddings. These patches are then applied to the encoder for generating sequence of contextual encodings (where specific relation of the patch is specified in a special context). These contain rich semantic information and can be utilized by the decoder.

It is observed that output embeddings correspond to respective image patches, further embeddings are utilized to obtain class labels with point-wise linear decoder or decoding using mask transformer. This decoder when pre-trained for classifying image shows that fine-tune can be done on datasets with moderate-size. Here, decoder is ready for semantic segmentation. The linear decoder then allows to achieve exceptional results; however, further performance improvement can be accomplished to generate class masks by mask transformer.

ii) Decoder: The mapping between patch-level (PL) encodings to PL class scores is achieved by point-wise linear decoder. Subsequently, these PL class scores are up sampled through bilinear interpolation to PL scores, and segmentation map is obtained from class dimension using softmax. Mask transformer, i.e. decoders, inserts set of learnable



(a) Patch size 32 × 32 (b) Patch size 16 × 16 (c) Patch size 8 × 8 (d) Ground Truth

Fig. 11 Description of ViT segmenter model [57]

class embeddings, wherein each class embeddings are randomly initialized and allocated to single semantic class. The masks are generated by mask transformer (i.e. scalar product among patch embeddings), which are entered to the decoder and class embeddings yield by decoder. Here, class masks are computed, further, final segmentation is achieved through softmax applied on class dimension, followed by LN to obtain class score (pixel-wise). Thus, it is verified that patch sizes are a key factor. The sharper boundaries can be obtained through patch size reduction, whereas incrementing patch size results in coarser image representation.

CNN-based model or CNN along with the transformer-based models basically split, class embeddings and pixel into two different streams due to the computational constraints. Herein, both are jointly processed during the decoding phase, and hence allow the production of dynamic filters with changing inputs. Subsequently, it is accepted as true that their encoder–decoder transformer in end-to-end manner, firstly, offers unified technique, for instance, segmentation, panoptic segmentation, and semantic segmentation.

This study aims to show that the transformer model in combination with other already existing models to make a great difference in the computational power and accuracy leads to a new research work and future directions.

You only look once (YOLO) [6]

It is noticed that YOLO achieved better results and outperformed other real-time OD algorithms with higher performance. When compared with other R-CNN object detection models, paradigm of proposal detection and verification is not followed by them. Instead, the proposed architecture uses end-to-end neural network, predicts bounding boxes along with class probabilities in one step, similar to fully convolutional networks. Figure 12 shows the application of objects detection using YOLO.

YOLO architecture consists of total 24 convolutional layers with two dense layers at output stage. It divides image into equal dimensional regions called grids. These corresponding grids predict the bounding boxes with respect to each cell's coordinates. This means that the model reframed OD as single regression problem, i.e. finding image pixels, obtaining coordinates of bounding box, and producing class probabilities. In addition, this model also greatly lowers computations, because detection followed by recognition is performed by the cells of image, unlike techniques such as region proposal and sliding window. The entire image scanning is performed by YOLO during train and test stage; therefore, it implicitly encoded the contextual

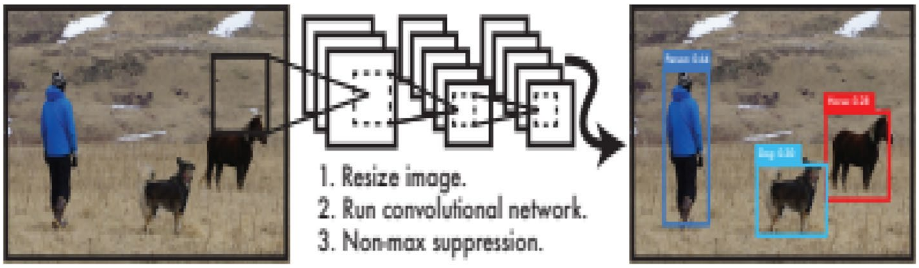


Fig. 12 YOLO [6]

information belonging to the classes and corresponding global appearances. YOLO design offers real-time speeds and training in end-to-end fashion.

From the past reviews about YOLO, it is concluded that YOLO is unified model to perform OD, ease in constructing and training can be accomplished on full image. Contrasting classifier-based techniques, YOLO's training achieves better performance due to loss function directly linked with detection performance, hence, entire model trained, jointly. Thus, YOLO suited ideal for robust and fast OD applications.

Over the years, varieties of improvements proposed on YOLO's v2 and v3 versions to enhance detection accuracy and focusing to maintain higher detection speed [30]. Despite the improvement in detection speed, we notice drop in localization accuracy, compared with two-stage detectors, especially for small objects. Lower localization accuracy witnessed by imposing strong spatial constraints on bounding box prediction, and hence struggled with small objects in groups. So, bounding boxes prediction from dataset is also reported by the model. Therefore, it struggles in object generalization when input object consists unusual aspect ratios or new object. The main source of error is the incorrect localization. When faster R-CNN is combined with YOLO, mAP is increased by 3.2%.

YOLOS [30]

This model is a transformer block like the YOLO CNN-based model. In addition to the ViT, the model consists of a detector portion of the network that maps a generated sequence of detection representations to class and box prediction. The ViT is developed to handle long-range dependencies wherein global contextual information is stored rather than region-level and local relations. Moreover, ViT does not support hierarchical architecture, whereas modern CNNs are capable of handling large variations in scale (input). From the literature, the point is unclear regarding pure ViT capability to transfer pre-trained visual representations belongs to image-level recognition task to complicated 2D OD task.

YOLOS is a simple with attention-only network and directly constructed upon ViT, it uses object query tokens (i.e. multiple learnable parameters) instead of class token. Here, bipartite matching loss is utilized for OD similar to DETR model. YOLOS exhibits, flexibility of ViTs in OD in learning features in sequence-to-sequence fashion along with optimal 2D inductive biases of image. Although being effective, YOLOS fails to remove CNN networks for high CV tasks. The performance of model is encouraging and preliminary results show the meaningful information, which suggests the robustness and generalization capability of transformer for varieties of downstream tasks. In comparison with other YOLO models, the accuracy is not best in class but has a future scope:

i)YOLOS needs 150 epochs for transfer learning (TL) to build pre-trained ViT for the task of OD, and performance in terms of detection accuracy is low, thus, scope for improvement is inherent.

ii)Focus is to work upon learning through visual representation for task-agnostic transformer models than task-oriented design with the fewest possible costs.

It is noted that we can feasibly combine other recent ViTs along with transformer-based detection heads and develop pure ViT network like DETRs, the VITFRCNN, the segmenter, and many more. Mostly, models for CV tasks are seen to replace or compute the same architecture in the transformer sector.

Here, Table 1 summarizes various models applied for the task of semantic segmentation, OD, and image classification based on model highlights and limitations, after investigating the studies on image classification, object detection, and image segmentation.

Tasks and model evolution

In this section, we discuss the task and model evolution for OD work.

Technical evolution of bounding box (BB) regressor

Bounding box (BB) regression is a vital method in OD wherein it refines the location of predicted BB through initial proposal or anchor box. The evolution of BB regression is as follows: absence of BB regression (before 2008), to from BB to BB (2008—2013), and then to from feature to BB (after 2013) (R-CNN, fast R-CNN, faster R-CNN, and YOLO).

After transformer-based models are proposed, DETR, VITFRCNN, and other object detection-specific models are based upon extracted features from the encoders. For fully transformer-based models, extracted features from the encoders are computed through decoders for predicting the boxes and sent through the prediction head (i.e. FNN).

For transformer-based models along with other detectors used, features are extracted from encoders for predicting boundary boxes, while some models even extract features from CNN and predict the boxes and select anchors using the decoders. The BB regressor method remains same "features to BB", where features are extracted from different modules or predicted through different models, and this depends on the architecture of the detector.

Evolution of non-max suppression (NMS)

Non-maximum suppression is one of the significant group of methods in OD. Due to similar scores of detection because of neighbouring windows, NMS employed the step of post-processing for—1) removal of duplicate BB, and 2) to obtain detection result. The NMS [59] is gradually advanced into three groups of techniques, discussed as follows:

(i) *Greedy Selection (GS):* It is applicable to overlapped detection of BB consisting maximum score of detection, whereas others are removed. This selection is used by R-CNN, fast R-CNN, faster R-CNN, and YOLO. The GS still considered the strongest baselines for today's OD purpose.

(ii) *BB Aggregation:* It utilizes full attention of object relationships and corresponding spatial layouts.

(iii) *Learning to NMS:* These showed encouraging results in enhancing the occlusion, dense OD over the traditional handcrafted NMS techniques.

It is noticed that transformer-based models like DETR do not need NMS in their design because of its set-based loss. It is predicted in [59] that learnable NMS techniques, relation models explicitly design the relations among different predictions along with attention. By utilizing direct set losses, no post-processing steps are required. Fully

Table 1 Summary of state-of-the-arts for semantic segmentation, OD, and image classification

Task	Method	Design	Highlights	Limitations
Image classification	ViT [46]	Encoder—NLP transformer for images	Transformer (global self-attention) applied on patches of image	Large-scale dataset training (image size—300 M)
		Linearly embedding—image patches by positional embedding	Convolution-free network	Careful TL for new task
			Outperforms ResNet	Large model consisting 632 M parameters for SOTA results
			It also attained excellent results while requiring substantially fewer computational resources for training purpose (i.e. almost four times in terms of computational efficiency and accuracy)	
Object detection	RCNN [1]	Resized and cropped regions classification by CNN	Frist real-time efficient object detection model using CNNs	Slow training and detection
		Region proposal BB refinement by SVM, trained by CNN features	Allows custom region proposal	More training time for classification of 2000 region proposals each image
				Selective search method is adopted, thus, no learning in the stage
				Generates bad region proposals
	Fast RCNN [2]	Fast R-CNN along with edge boxes applied for region proposals generation	R-CNN must classify every region	Fast R-CNN performance is low due to region proposals identification
		R-CNN used for cropping and resizing region proposals	Fast R-CNN pools features from CNN belongs region proposal	Fast R-CNN is good when not region proposals. Therefore, estimating region proposals
		Fast R-CNN handles entire image	Fast R-CNN efficiently worked compared to R-CNN, because, estimations are shared for overlapping regions	
	Faster RCNN [3]	No selective search method	Optimal run-time performance	RPN training is performed with all anchors (mini-batch—size 256)
		Separate network for predicting region proposals	Improvement over its predecessor with respect to run-time speed and raw performance	Extraction perform on single image

Table 1 (continued)

Task	Method	Design	Highlights	Limitations
		Region proposals prediction reshaped by layer of RoI pooling, later used to classify image under proposed region, further, prediction of offset for BB	RPN is faster as compared to Selective Search	Network convergence is slow
			Fixed-length features are extracted from every region proposal by layer of RoI pooling	
	DETR [28]	Linear projection layer is applied for CNN feature dimension reduction	End-to-end pipeline of training using transformer for OD	More time for convergence
		Encoder–decoder consists of spatial positional embedding at each layer of multi-head self-attention	No manual post-processing stage	Low detection accuracy for small objects
		Output positional encoding, i.e. object queries, is added to the layer of multi-head self-attention in decoder		
		Hungarian loss is used		
	D-DETR [29]	Deformable transformer with deformable attention layers for sparse priors	Better performance for small object compared to DETR	SOTA results using 52.3 AP
		Applied multi-scale attention	Converged fast compared to DETR	Augmentation in test time
	VITFRCNN [18]	Transformers for encoding visual features while RPN for detecting outputs	Pre-training capacity is large	More training time on large-scale dataset (300 M)
		Adds detection network to ViT	Fine-tuning performance is fast	Training from scratch is difficult for smaller datasets
		ViT used for state related to input class token and is outputted through MLP classification head	Investigated improvements—superior performance is reported for image in out-of-domain, and better performance for large objects	GPU memory limitation
			Avoiding spurious over detections	Self-attention and convolutional layers relationship, and limitations of CNNs
	YOLO [6]	YOLO predicts the BB	YOLO is faster than other OD algorithms	Small objects detection issue
		YOLO estimates the class probabilities for BB	Better accuracy of prediction and better IoU in BB	Spatial constraints lower small objects detection

Table 1 (continued)

Task	Method	Design	Highlights	Limitations
	YOLOS [59]	Transformer block like YOLO CNN-based model	2D OD is accomplished in pure sequence-to-sequence way with optimal addition of inductive biases	150 epochs needed for TL
		In addition to ViT, this model consists of detector portion of the network that maps a generated sequence of detection representations to class and box prediction	Performance is encouraging	Learning through visual representation
			Preliminary outcomes are significant	
	Rank-DETR[70]	Rank-based design with	SOTA is improve by Rank-DETR	Rank-based design needs to be explored more
		prompt engineering	Backbone of ResNet-50, Swin-L, and Swin-T is used for enhancing localization accuracy	More computing time
		Rank-based loss calculation, matching cost for accurate localization accuracy rank	More AP under higher IoU	
Semantic segmentation	Mask RCNN [8]	Mask R-CNN used for predicting an object mask (RoI), also recognized BB	Simple to train and outperforms state-of-the-arts	Process still images, not capable to explore temporal details of object
		Perform image segmentation, i.e. Semantic and Instance	Contains small overhead compared to faster R-CNN	Fails in detecting object suffering from low resolution
			Generalization is easy	
	ViT Segmenter [57]	Encoder: projecting image patches in sequential embeddings, further, encoding using transformer	Global context captured by transformer	Not computationally feasible
		Decoder: mask transformer result from encoder, class embeddings, further, predicts segmentation masks	Decoder: simple point-wise linear is applied to patch encodings for better results	Reduction in patch size requires the computation of attention along longer sequences
			Unified model for semantic and instance segmentation, and segmentation of panoptic	More computing time

transformer-based models already consist of inbuilt functionalities needed for implementing NMS.

Technical evolution of hard-negative mining (HNM)

In [3], imbalanced data issue during training is investigated. Technical evolution related to HNM in OD like bootstrap in OD refers to group of training methods wherein small part belonging to background samples is considered for training. Further, it iteratively added new misclassified backgrounds during training, without HNM. Later, during the deep learning era with improvement in computational power, faster R-CNN and YOLO easily balanced weights among positive and negative windows. These further improvements led to bootstrapping with new loss function.

DETR transformer models are equivalent to faster R-CNN training process, which balances the obtained proposals of positive or negative by subsampling [60]. Moreover, matching cost is independent object prediction. When transformer models are used in conjunction with other R-CNN models, they follow their previous HNMs.

Applications of transformers

Transformer-based models are emerged as a competitive alternative to CNN models on OD. The use of transformers-based learning for visual representation developed sparked interest in the CV community. In [43], authors reviewed some of the important detection applications from the past few years in topics such as face detection, traffic sign detection, and pedestrian detection. Here, discussion is also focused on the difficulties and challenges faced in each area along with certain object detectors (which are mostly CNN-based) that are able to resolve the challenges.

In this section, the same issues are discussed, but transformer-based proposed models help to overcome the challenges faced by them and then propose a few new signs of progress that are achieved in the same areas.

Pedestrian detection

In one of the important object detection applications, i.e. pedestrian detection is applied in scene perception and object detection in autonomous vehicles, criminal investigation, video surveillance, etc. In general, DL-based OD methods are greatly progressing in this field and are constantly been improving to face number of challenges, gaining accuracy and overcoming issues. In a real-time-based detection application, it is made sure that the model should compute the best results at all times.

DL ODs such as fast/faster R-CNN presented their best performances for general detection; however, it suffers from limited achievement in detecting small pedestrians under low resolution of convolutional features. Recent solutions are proposed which helps to improve and add specific features to overcome this issue. There is still requirement to enhance hard-negative detection because certain background image patches are exactly similar to pedestrians in corresponding visual appearance. Features in deeper layers of CNN consist of better semantics, but not qualified to detect dense objects and are a reason for occlusion. The CNN utilized successfully in pedestrian detection [61] and achieved the promising outcomes.

It is noted that the fine-tuning process improves generalization and till date, transformer networks are tested as backbones and found out to be outperforming CNNs in terms of generalization and absorbing large-scale datasets for learning robust representation. End-to-end detectors, DETR, and deformable DETR accomplish comparatively better results for common OD. Hence, due to its unique model architecture and competitive results with fast R-CNN models, it is applied to pedestrian detection.

To make DETR practically possible and even for crowded detection, new decoder with dense queries and rectified attention unit, i.e. DQRF, is introduced, which is easily implementable and benefits to alleviate identified problems of DETR in detection of pedestrian. In [34], faster matching algorithm using bipartite scheme is proposed wherein improvements are suggested for DETR pertaining to annotations of visible box. Further, Rank-DETR [70] is designed to predict the high-quality OD based on the rank of bounding box which accurately identifies the positive prediction and subdue the negative predictions. This approach achieved higher localization accuracy by enhancing AP.

Number of models can be made in conjunction with attention-based or transformer-based encoders or decoder modules with already existing object detectors. For example, in [62], PedesFormer is a swim transformer-based model that focuses on the advancement of research. Segmentation and domain adaptation are constructed using UNet network with swim transformer, as it can be applied spatial constraints to the pedestrian detection [13]. Similar to this approach, many models are seen to perform the same task with more improvement. A combination of object detection models with semantic tasks is able to put to end the hard-negative samples. Likewise, transformer-based semantic models can also be applied to achieve same or better results with additional benefits.

Face detection

Detection of face is the oldest task in CV and is continuously grown and evolved since then. It is now applied to many other tasks. Due to rapid progress of DL in CV, many DL-based frameworks are developed for detecting the face, periodically, which achieved improvements in accuracy. In DL era, mostly, face detection methods follow detection idea of general ODs such as faster R-CNN [3] and SSD [5].

Some of the issues and challenges to detect face are identified as follows:

(i) *Intra-Class Variation*: We know that varieties of skin colours, expressions, movements, and poses are possible from Human faces.

(ii) *Occlusion*: It may happen that faces are partly occluded by another objects.

(iii) *Multi-Scale Detection*: Face detection from range of large variety, especially when tiny faces present good encounters.

CNN is based up on the local ideas for feature expression, which result in low efficiency to capture long-range pixels dependency, thus, poor performance is reported to recognize facial expression. Further, to overcome this problem, self-attention approach is added using residual network, due to which recognition is done at global level. CNN is not capable to perform the encoding of different features in relative position, while attention models can learn different features with focus on the interactive ones. Some of the published researches explained about the models, which are experimented with the combination of attention models with already existing detectors in order to improve results. Here, encoding using transformer is used in capturing relationships amid various action units aimed at expressions in wide range in training dataset and thus yields high classification performance. It is seen that transformers are robust in perturbations, domain shifts, occlusions; hence with this view, TFE model is developed in [63]. Similar to this, many transformer-based modules are grouped with already existing object detectors, where the basic feature and working for the modules remain the same. Though, architecture and improvements vary in the process of detection.

Traffic signal detection

In recent past, detecting lights and traffic signs automatically attracted the researchers due to advancement in self-driving technology. It is also mistakenly seeming to assume the recognition of traffic signs is simple, there are lot of challenges present. This detention would be practically difficult on driving under different conditions such as night and sun glare. So, image capturing through camera may be blurred due to car in motion. During bad weather conditions, it becomes even impossible to detect the image. However, there is a need of real-time detection for autonomous vehicles with prominent accuracy. Since in this deep learning era, models like faster R-CNN and SSD are also applied in traffic sign detection. To overcome certain drawbacks, new techniques are continuously developed and implemented.

It is observed that comparatively higher accuracy is obtained by the two-stage algorithms, however, recognition speed is slow. Therefore, attention modules can be utilized with other CNN models to overcome the difficulties [64]. Transformer-based models can also be used in conjunction with CNN models, the self-attention module and multi-headed attention have the capabilities to recognize the image, both locally and globally, and that leads to further scope of future research.

General application

Therefore, transformers showed the promising results on training models with multi-modal input data. It avoided the heavy engineering and inefficiencies during utilization of mixed architectures. For the machines to become more useful, algorithms should learn how to aim with multi-sensory inputs. Transformer should be perfect and it is expected to be applied on a large scale for different applications in the near future. It is a fascinating cause to advance as a multi-task, and can easily be tailored into already existing models in order to add new features. ViTs are now applied in 3D analysis, video processes, and generative modelling.

Datasets and evaluation metrics

Datasets

For the purpose of object detection, various datasets are publically available for the performance evaluation. There exists a wealth of readily available, open-source datasets that can be harnessed for experimentation and model development. Some datasets are listed as follows;

1. *CIFAR-10* [65]: CIFAR-10 is a comprehensive dataset that consists of 60,000 colour images in 10 different categories. The dataset holds 10,000 test images and 50,000 training images split into five training groups. However, the images in CIFAR-10 are low resolution (32 × 32), thus, dataset allows researchers to quickly apply different algorithms.
2. *Open Images* [66]: Open Images V4 dataset offers large scale across several dimensions wherein 30.1 M image-level labels for 19.8 k concepts, 15.4 M bounding boxes for 600 object classes, and 375 k visual relationship annotations involving 57 classes are observed. Specifically for object detection, 15 × more bounding boxes than the next largest datasets (15.4 M boxes on 1.9 M images) are provided. The images often show complex scenes with several objects (eight annotated objects per image on average). Open Images V4 dataset competition also uses mean average precision (mAP) over the 500 classes to evaluate the object detection task.
3. *COCO Dataset* [41]: Common objects in context (COCO) dataset, a seminal resource, offers a diverse collection of images with objects situated in complex, real-world contexts. Its wide adoption stems from its capacity to evaluate object detection models across intricate scenarios. In COCO, there are more small objects than large objects. Specifically, approximately 41% of objects are small ($\text{area} < 322$), 34% are medium ($322 < \text{area} < 962$), and 24% are large ($\text{area} > 962$). The general average precision (AP) and average recall (AR) are averaged over multiple Intersection over Union (IoU) values. Specifically, we use 10 IoU thresholds of 0.50:0.05:0.95.
4. *PASCAL VOC* [40]: PASCAL VOC, albeit smaller in scale than COCO, has played a pivotal role in benchmarking object detection algorithms. Its twenty object classes have made it a valuable asset for early stage evaluations. The current metrics used by the current PASCAL VOC object detection challenge are the precision × recall curve and average precision.
5. *ImageNet* [42]: ImageNet is initially devised for image classification, and then extended to encompass object detection challenges. It boasts an extensive array of annotated images, offering an invaluable resource for object detection researchers. ImageNet object localization challenge, which evaluates the performance of object localization algorithms, uses a specific error metric that considers both the class label and overlapping region between the ground truth and detected bounding boxes for each image. This metric calculates a "min error" for each image, indicating how well the predicted bounding boxes align with the true objects in the image.

Other datasets such as CIFAR-10, PASCAL VOC, and ImageNet have their significance in computer vision tasks, but COCO's extensive challenging dataset, its real-world complexity, and rich annotations have gained prominence as a benchmarking for state-of-the-art object detection models.

Existing datasets provide a strong starting point; moreover, we can always have an option to build custom datasets. By creating custom datasets, one can curate a collection of images and annotations that mirror the real-world scenarios and challenges encountered in the target application.

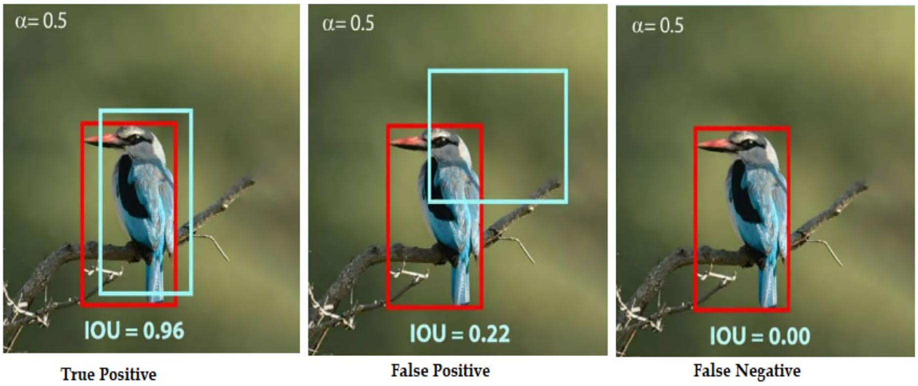


Fig. 13 Object detection prediction

Evaluation metrics

Mean average precision (mAP), common metric utilized in evaluating the accuracy of OD models. It gives a detailed overview of how a model is performing alongside its competitor models on the same dataset. Here, metrics involved in mAP are discussed, i.e. confusion matrix (CM), Intersection over Union (IoU), precision, and recall [67].

Intersection Over Union: A number that quantifies the degree of overlap between two boxes. Here, in OD and segmentation, an IoU estimating overlaps related to prediction region and ground truth (GT). However, IoU is considered as primary metric for segmentation to report model accuracy.

Using IoU threshold value, the prediction of true positive (TP), false negative (FN), or false positive (FP) can be decided, collectively known as confusion matrix. Figure 13 shows IoU for the detection of object in the image.

Confusion Matrix: A CM is a table that shows performance of a classifier given some truth values/instances.

True positive—classifier predicted positive wherein truth is positive, false positive—wrongly predicting positive, i.e. IoU < α (for detection), false negative—no detection by classifier, and true negative—correct prediction for negative class.

Similarly, in segmentation and OD the exact words are not same (see Fig. 14). In OD, correctness of estimate (TP, FN, or FP) is confirmed with IoU threshold, whereas in segmentation, this is decided through referring GT pixels.

Precision: The total TP from total detections, i.e. TP and FP, this measure helps in identifying the positives prediction that is correct. If you are wondering how to calculate precision, it is simply the true positives out of total detections.

Recall: The total TP from total detections, i.e. TP and FN, helps in addressing the question of "What quantity of TP, predicted correctly?".

The average precision (AP) is considered as area under precision–recall curve, and it is calculated class-wise. The mAP is averaged over AP for all detected classes.

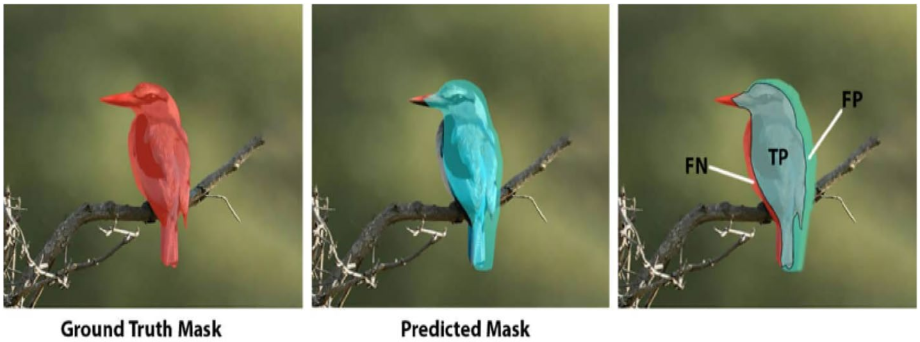


Fig. 14 Image segmentation prediction

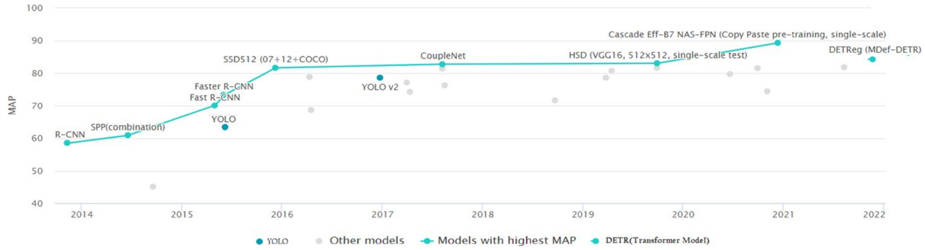


Fig. 15 OD on PASCAL VOC 2007 at a universal level

Performance analysis and discussion

The COCO dataset (used for COCO test-dev and COCO minival benchmarks) and the PASCAL VOC 2007 dataset stand out as two pivotal datasets in the field of object detection. Over time, object detection models have evolved and improved significantly, with these datasets serving as key driving forces behind the advancements.

One comparative analysis on the current scenario of the models with the highest mAP along the varied timeline presented in Fig. 15 uses PASCAL VOC 2007 dataset. This shows how transformer model (DETReg) tried to reach the mark better than CNNs. Further, Fig. 16 shows COCO 2017 dataset to evaluate the performance of different object detection models, here, ranked Co-DETR [68] has achieved higher box mAP of 66.0%. Moreover, the performance of different object detection models on MS-COCO, VOC07, and VOC12 datasets being previously indicated in Fig. 2 witnessed OD performance improvement (see Sect. "Object detection models"). Moreover, from Fig. 17 we can depict that ranked Co-DETR [68] offers the higher box average precision of 65.9% for OD task.

On the MS-COCO dataset which is based on the average precision, the best real-time OD algorithm until September 2022 was YOLOv7, followed by vision transformer such as Swin and DualSwin, PP-YOLOE, YOLOR, YOLOv4, and EfficientDet. As of July 2023, an evolving trend in the realm of real-time OD on the MS-COCO dataset is the ascent of Co-DETR, marking a significant shift in the landscape.

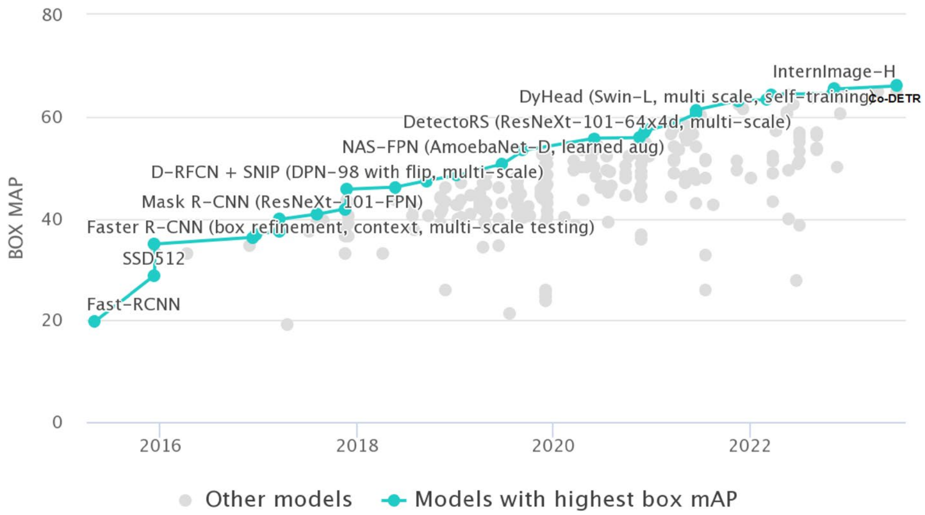


Fig. 16 OD on COCO 2017 test-dev at a universal level [68]

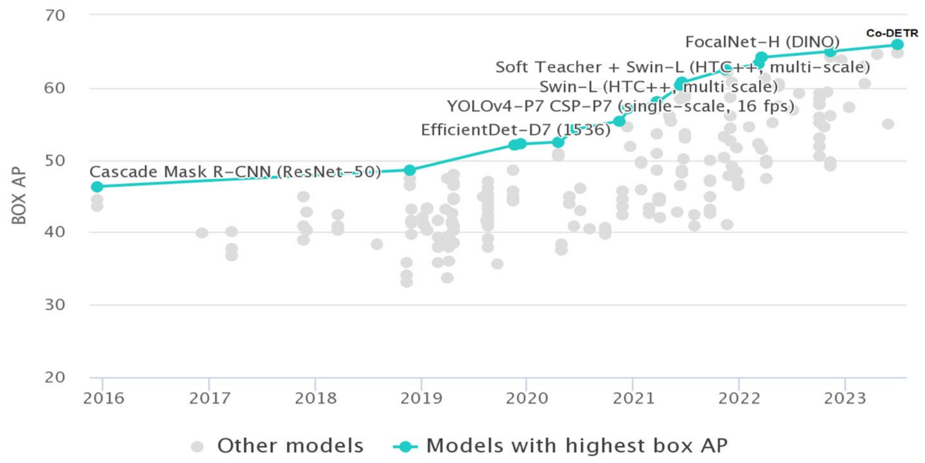


Fig. 17 OD on COCO 2017 minival at a universal level [68]

Table 2 provides a valuable tool for comparing and assessing various models across different datasets and evaluation metrics using quantifiable numerical data. Here, we can determine and decide which model is more suitable for different computer vision tasks, taking into account the specific requirements and challenges posed by each dataset. The image classification task using ViT-L model on ImageNet achieved a higher accuracy of 87.76%. Subsequently, OD task reported the highest mAP of 73.20% on PASCAL VOC using fast R-CNN.

Challenges in object detection

Following are the challenges in object detection;

1. *Small objection detection* Large object detection is performed accurately by the various object detectors but deprived performance is reported on small objects.

Table 2 Comprehensive evaluation metrics and dataset comparative analysis for CV models

Reference	Task	Model Used	Dataset	Results
[1]	Object detection	RCNN	PASCAL VOC	mAP—58.50%
[2]	Object detection	Fast RCNN	PASCAL VOC	mAP—70%
			COCO 2017 test-dev	Box mAP—19.7
				mAP—19.70%
[3]	Object detection	Faster RCNN	PASCAL VOC	mAP—73.20%
			COCO 2017 test-dev	Frame per secs—46.7
				mAP—21.90%
				Average mAP—16.4
[28]	Object detection	DETR	COCO 2017	Average precision (AP)—43.0
				Average mAP—17.7
[29]	Object detection	D-DETR	COCO 2017	Average precision (AP)—46.9
				Average mAP—18.5
				mAP—52.30%
[18]	Object detection	ViT-B/16-FRCNN	COCO 2017	Average precision (AP)—37.8
			OBJECTNET-D	Average precision (AP)—22.9
[6]	Object detection	YOLO	PASCAL VOC	mAP—63.40%
				Frame per secs—46.7
			COCO 2017	Average mAP—32
				mAP—43.50%
[30]	Object detection	YOLOS(VIT-B)	COCO 2017	Average mAP—20
[8]	Object detection	Mask RCNN	COCO 2017	Average mAP—17.6
				Box mAP(Real time)—45.7
[69]	Object detection	ResNet-101	Pascal VOC	Average mAP—63.7
[70]	Object detection	Rank-DETR (ResNet50)	COCO 2017	Average mAP—50.2
[46]	Image classification	ViT-H	ImageNet	Accuracy (Top 1)—88.55%
			CIFAR-10	Percentage correct—99.9%
[46]	Image classification	ViT-B	ImageNet	Accuracy(Top1)—85.2%
[46]	Image classification	ViT-L	ImageNet	Accuracy (Top 1)—87.76%
			CIFAR-10	Percentage correct—99.42%
			PASCAL VOC	mIoU—68
[57]	Semantic segmentation	ViT segmenter	PASCAL VOC	mIoU—59

2. *Multi-resolution* Object detection algorithms achieve good results under controlled environment for images in specific resolution. However, these algorithms disappoint on varied resolution of inputs.
3. *Large dataset* Applying CNN and transformer-based algorithms needs big datasets with suitable annotations which is laborious task. Further, various images are generated by the numerous resources to investigate useful information.
4. *Computational resources* To train the object detectors on big datasets requires the more computational power [58].

5. *Imbalance class* Images class imbalance pertaining to background and foreground images leads to lower model performance.
6. *Localization* To localize the objects and further perform the prediction, background pixels restrict the accurate prediction, thus, localization errors need to be reduced.

Conclusion and future scope

In the task of object detection, various object detectors using CNN-based model and transformer-based models are proposed in the literature. We have investigated the various domains in which object detection in real time is very important and needs substantial improvement. CNN-based object detectors lack in generalization and lower localization accuracy, whereas transformer-based detectors achieve higher detection accuracy and more generalization. It is observed that transformer models sparked the great interest in field of computer vision. One of the great benefits is their inclination towards building universal model architectures that can support any type of input data such as text, image, audio, and video. In this paper, a suitable explanation of the different transformer-based object detectors is demonstrated. In addition, the characteristics of each feature aims to give an idea of how transformers are able to flexibly fuse and conjunct with other DL models to improve on the efficiency. Thus, we suggest the need for new architectures and ideas for future research in particular OD and other CV tasks. Here, we offer a meaningful review to depict a line of difference between transformer-based detectors and CNN models. The utilization of transformer-based detector uses the power of existing DL models and various attention mechanisms to achieve the higher generalization which makes it more suitable for real-time objection detection.

Moreover, we covered the applications of transformers in CV, particularly in tasks of recognition such as segmentation, OD, and image classification. However, all these applications will be the next step to understand problems in-depth and to be motivated for modelling a new and robust architecture. Further, object detection can be explored in more fields such as multi-modal tasks, video processing, video forecasting, image super-resolution, and 3D analysis.

Author contributions

S. Shah performed the critical analysis on vision transformer and prepared the initial draft of manuscript. J Tembhurne examined and suggested the changes to review on the applications of vision transformer. All authors reviewed and finalized the manuscript.

Funding

No funding was received for conducting this study.

Data availability

All data generated or analysed during this study are included in this published article.

Declarations

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Received: 16 May 2023 Accepted: 6 November 2023
Published online: 20 November 2023

References

1. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
2. Girshick RJCS (2015) Fast R-CNN. arXiv preprint arXiv:1504.08083
3. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, vol 28
4. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, vol 29
5. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision. Springer, Cham, pp 21–37
6. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
7. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
8. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
9. Jiang H, Learned-Miller E (2017) Face detection with the faster R-CNN. In: 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). IEEE, pp 650–657
10. Martinson E, Yalla V (2016) Real-time human detection for robots using CNN with a feature-based layered pre-filter. In: 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE, pp. 1120–1125
11. Stewart R, Andriluka M, Ng AY (2016) End-to-end people detection in crowded scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2325–2333
12. Boujemaa KS, Berrada I, Bouhoute A, Boubouh K (2017) Traffic sign recognition using convolutional neural networks. In: 2017 International conference on wireless networks and mobile communications (WINCOM). IEEE, pp. 1–6
13. Zhang J, Liu C, Wang B, Chen C, He J, Zhou Y, Li J (2022) An infrared pedestrian detection method based on segmentation and domain adaptation learning. Comput Electr Eng 99:107781
14. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware CNN model. In: Proceedings of the IEEE international conference on computer vision, pp 1134–1142
15. Hafiz AM, Bhat GM (2020) A survey on instance segmentation: state of the art. Int J Multimed Inf Retr 9(3):171–189
16. Ansari MA, Kurchaniya D, Dixit M (2017) A comprehensive analysis of image edge detection techniques. Int J Multimed Ubiquitous Eng 12(11):1–12
17. Peng X, Schmid C (2016) Multi-region two-stream R-CNN for action detection. In: European conference on computer vision. Springer, Cham, pp 744–759
18. Beal J, Kim E, Tzeng E, Park DH, Zhai A, Kislyuk D (2020) Toward transformer-based object detection. arXiv preprint arXiv:2012.09958
19. Wang W, Xie E, Li X, Fan DP, Song K, Liang D, Shao L (2021) Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 568–578
20. Chu X, Tian Z, Wang Y, Zhang B, Ren H, Wei X, Shen C (2021) Twins: revisiting the design of spatial attention in vision transformers. Adv Neural Inf Process Syst 34:9355–9366
21. Xu W, Xu Y, Chang T, Tu Z (2021) Co-scale conv-attentional image transformers. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9981–9990
22. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10012–10022
23. Wu H, Xiao B, Codella N, Liu M, Dai X, Yuan L, Zhang L (2021) CVT: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 22–31
24. Huang Z, Ben Y, Luo G, Cheng P, Yu G, Fu B (2021) Shuffle transformer: rethinking spatial shuffle for vision transformer. arXiv preprint arXiv:2106.03650
25. Wang W, Yao L, Chen L, Lin B, Cai D, He X, Liu W (2021) Crossformer: a versatile vision transformer hinging on cross-scale attention. arXiv preprint arXiv:2108.00154
26. Chen CF, Panda R, Fan Q (2021) Regionvit: regional-to-local attention for vision transformers. arXiv preprint arXiv:2106.02689
27. Yang J, Li C, Zhang P, Dai X, Xiao B, Yuan L, Gao J (2021) Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641
28. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020). End-to-end object detection with transformers. In: European conference on computer vision. Springer, Cham, pp 213–229
29. Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable DETR: deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159
30. Fang Y, Liao B, Wang X, Fang J, Qi J, Wu R, Liu W (2021) You only look at one sequence: rethinking transformer in vision through object detection. Adv Neural Inf Process Syst 34:26183–26197
31. Ebrahimpour R, Kabir E, Yousefi MR (2007) Face detection using mixture of MLP experts. Neural Process Lett 26(1):69–82
32. Kim B, Lee J, Kang J, Kim ES, Kim HJ (2021) HOTR: end-to-end human-object interaction detection with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 74–83
33. Li M, Han D, Li D, Liu H, Chang CC (2022) MFVT: an anomaly traffic detection method merging feature fusion network and vision transformer architecture. EURASIP J Wireless Commun Netw 2022(1):1–22

34. Lin M, Li C, Bu X, Sun M, Lin C, Yan J, Deng Z (2020) DETR for crowd pedestrian detection. arXiv preprint arXiv:2012.06785
35. Song H, Sun D, Chun S, Jampani V, Han D, Heo B, Yang MH (2022) An extendable, efficient and effective transformer-based object detector. arXiv preprint arXiv:2204.07962
36. Meinhardt T, Kirillov A, Leal-Taixe L, Feichtenhofer C (2022) Trackformer: multi-object tracking with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8844–8854
37. Wang Y, Xu Z, Wang X, Shen C, Cheng B, Shen H, Xia H (2021) End-to-end video instance segmentation with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8741–8750
38. Wang Y, Zhang X, Yang T, Sun J (2022) Anchor DETR: query design for transformer-based detector. In: Proceedings of the AAAI conference on artificial intelligence, vol 36, No 3, pp 2567–2575
39. https://odsc.medium.com/vision-transformer-and-its-applications-265a629c0cf4. Accessed 20 Dec 2022
40. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
41. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Zitnick CL (2014). Microsoft coco: common objects in context. In: European conference on computer vision. Springer, Cham, pp 740–755
42. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255.
43. Zou Z, Shi Z, Guo Y, Ye J (2019) Object detection in 20 years: a survey. arXiv preprint arXiv:1905.05055
44. Lindeberg T (2012) Scale invariant feature transform, 10491
45. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 886–893
46. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Houlsby N (2020) An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
47. Li J, Wei Y, Liang X, Dong J, Xu T, Feng J, Yan S (2016) Attentive contexts for object detection. IEEE Trans Multimed 19(5):944–954
48. Khan S, Naseer M, Hayat M, Zamir SW, Khan FS, Shah M (2022) Transformers in vision: a survey. ACM Comput Surv 54(10s):1–41
49. Hu H, Gu J, Zhang Z, Dai J, Wei Y (2018) Relation networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3588–3597
50. Tolstikhin IO, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, Dosovitskiy A (2021) MLP-mixer: an all-MLP architecture for vision. Adv Neural Inf Process Syst 34:24261–24272
51. https://keras.io/examples/vision/object_detection_using_vision_transformer/. Accessed 22 Dec 2022
52. Lin T, Wang Y, Liu X, Qiu X (2022) A survey of transformers. AI Open
53. Chi C, Wei F, Hu H (2020) Relationnet++: bridging visual representations for object detection via transformer decoder. Adv Neural Inf Process Syst 33:13564–13574
54. Long J, Shelhamer E, Darrell T, Berkeley UC (2014) Fully convolutional networks for semantic segmentation. arXiv preprint arXiv:1411.4038
55. Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, Torr PHS (2020) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. arXiv preprint arXiv:2012.15840
56. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P (2021) SegFormer: simple and efficient design for semantic segmentation with transformers. Adv Neural Inf Process Syst 34:12077–12090
57. Strudel R, Garcia R, Laptev I, Schmid C (2021) Segmenter: transformer for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7262–7272
58. Diwan T, Anirudh G, Tembhurne JV (2022) Object detection using YOLO: challenges, architectural successors, datasets and applications. Multimed Tools Appl 82:1–33
59. Hosang J, Benenson R, Schiele B (2017) Learning non-maximum suppression. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4507–4515
60. Canévet O, Fleuret F (2015). Efficient sample mining for object detection. In: Asian conference on machine learning. PMLR, pp 48–63
61. Xu Z, Li B, Yuan Y, Dang A (2020) Beta R-CNN: looking into pedestrian detection from another perspective. Adv Neural Inf Process Syst 33:19953–19963
62. Hasan I, Liao S, Li J, Akram SU, Shao L (2022) Pedestrian detection: domain generalization, CNNs, transformers and beyond. arXiv preprint arXiv:2201.03176
63. Jacob GM, Stenger B (2021) Facial action unit detection with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7680–7689
64. Huang H, Liang Q, Luo D, Lee DH (2022) Attention-enhanced one-stage algorithm for traffic sign detection and recognition. J Sens 2022:3705256
65. Doon R, Kumar Rawat T, Gautam S (2018) Cifar-10 classification using deep convolutional neural network. In: 2018 IEEE Punecon, Pune, India, pp 1–5. https://doi.org/10.1109/PUNECON.2018.8745428
66. Kuznetsova A, Rom H, Alldrin N, Uijlings J, Krasin I, Pont-Tuset J, Ferrari V (2020) The open images dataset v4: unified image classification, object detection, and visual relationship detection at scale. Int J Comput Vis 128(7):1956–1981
67. Padilla R, Netto SL, Da Silva EA (2020) A survey on performance metrics for object-detection algorithms. In: 2020 International conference on systems, signals and image processing (IWSSIP). IEEE, pp 237–242
68. Zong Z, Song G, Liu Y (2023) DETRs with collaborative hybrid assignments training. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6748–6758
69. Krishna O, Ohashi H, Sinha S (2023) MILA: memory-based instance-level adaptation for cross-domain object detection. arXiv preprint arXiv:2309.01086

70. Pu Y, Liang W, Hao Y, Yuan Y, Yang Y, Zhang C, Huang G (2023) Rank-DETR for high quality object detection. arXiv preprint arXiv:2310.08854

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.