Object Detection Using Vision Transformers - An Overview

Electronic Engineering

Project Work

submitted by
Abdul-Azeez Olanlokun

Electronic Engineering
Mat.Nr.: 2180593
<abdul-azeez.olanlokun@stud.hshl.de>

December 3, 2023

First supervisor:	Prof. Dr. Stefan Henkler
Second supervisor:	Prof. Dr. Achim Rettberg

Contents

1 Introduction

Object detection is a fundamental task in computer vision that is critical in a wide range of applications, from autonomous vehicles to medical imaging and security monitoring. Traditionally, object detection has been dominated by convolutional neural networks (CNNs), which have shown remarkable success in localizing and classifying objects in images. However, a new breakthrough in the field of computer vision has emerged with the introduction of Vision Transformers (ViTs), a unique design that takes the power of transformers into the world of image analysis [VSP+17]. Transformers, which were initially designed for natural language processing tasks, have demonstrated amazing capabilities for modeling long-term dependencies in data, making them a compelling option for vision tasks. The move from CNNs to Vision Transformers has prompted tremendous interest and investigation in the area, with academics and practitioners attempting to capitalize on the prospective of these architectures for object detection [DBK+21]. The focus of Vision Transformers is shifting away from grid-like convolutions and toward self-attention mechanisms, which enable global context awareness and enhanced object detection. This transformation provides a new viewpoint on object detection, allowing for more precise, efficient, and adaptable solutions. The theoretical foundations, practical implementations, and benchmark performance of ViT-based object detection models are discussed, as well as key considerations in adapting transformers for this task, such as positional encodings, input data formats, and architecture variations [TCD+21]. This paper is an overview, which explores the exciting frontier of object detection using Vision Transformers, examining the theoretical foundations, practical implementations, and benchmark performance. It discusses key considerations in adapting transformers for this task, such as positional encodings, input data formats, and architecture variations, and delves into prominent ViT-based object detection models such as DETR [CMS+20].

1.1 Motivation

For many natural language processing (NLP) applications, the Transformer [VSP+17] model has emerged as the go-to option. It has demonstrated remarkable advancements in text categorization [RSR+20], machine translation [LLG+19], question answering [DCLT18], document summarization [ZWZ19], and other areas. This success can be attributed in part to the Transformer's scalability, which allows for the pretraining of models of remarkable size on large datasets without showing any signs of saturating performance, and its ability to learn complex dependencies between input sequences via self-attention [DCLT18, RNSS18, RWC+19, BMR+20].

The first example of a transformer architecture being directly applied to an image was provided by the Vision Transformer (ViT) [DBK+21], which treated an image as a sequence of patches. While it still performs less well than convolution-based models on mid-sized

datasets, the ViT appears to have retained the capability of NLP transformers, allowing it to pre-train on an unparalleled volume of data. Essentially, ViT raises the possibility of adding or substituting attention-based components for the usual convolution, which has been the cornerstone of vision modeling for many years. The vision transformer [DBK+21] (ViT) is a straightforward transformer adaptation for computer vision applications like image classification. It works by splitting the input picture into non-overlapping patches, which are then fed to a vanilla transformer architecture after a linear patch projection layer. Transformers provide parallel processing and a whole field of view in a single layer, in contrast to networks constructed from convolutional layers. Transformers, along with other attention-based designs, see e.g. [BZV+19, CMS+20], have had a significant impact on computer vision architecture design lately. A lot of contemporary computer vision architectures either directly borrow elements of their design from this work, or are at the very least motivated by the latest discoveries made possible by transformers [CMS+20, DBK+21, TCD+21]. Many computer vision tasks, such as object detection and segmentation [ATC+21], video analysis [ADH+21, FXM+21], and picture synthesis [CZJ+22, HZ21], have advanced significantly as a result.

1.2 Goals

This paper aims to give an overview of object detection using vision transformers comprehensively. The fundamental purpose of adding Vision Transformers in object detection is to increase the accuracy of object localization and classification in images. ViTs seek to deliver more precise and dependable outcomes in a wide range of applications by using self-attention processes and global context comprehension. With this goal to enhance object detection accuracy, this paper aims to explore the fundamentals of vision transformers in object detection, its process, and methods in making sure object detection is improved in the area of computer vision. Furthermore, we aim to evaluate and compare different object detection methods such as traditional CNNs with our ViT's Model;DETR etc, by analyzing and comparing them on the same Datasets, such as COCO. We want to identify the strengths and drawbacks of ViTs in object detection in the domain of computer vision. This paper's contributions include identifying the challenges faced by ViTs for object detection, and to suggest ways to enhance its capabilities. Additionally, we aim to provide a better overview on the essential objective which is to have a better understanding of how Vision Transformers make decisions and to improve their interpretability. This will aid in the development of confidence in ViT-based object detection systems, especially in applications where safety and dependability are crucial.

1.3 Overview

This paper is an overview of the works on object detection using vision transformers. To be able to understand comprehensively, the objectives and goals of vision transformers for object detection in computer vision applications, where object detection has been of great interest in recent years, especially in the area of autonomous driving, it is important to know the impacts of vision transformers.

This paper has been arranged into six chapters to address these impacts. Chapter two2

highlights relevant research in the topic, which includes prior studies' successes and future aspirations. This chapter reviews relevant works in classic CNN-based object detection and the growing area of ViT-based object detection.

Chapter three3.1 comprehensively evaluates the main structure, which is the Transformer, its architecture, and also key features of the transformer, such as positional encoding and self-attention mechanism etc.

Chapter four4 presents our main focus, which is the Vision Transformer for object detection. This chapter comprehensively analyzes the structure, ViT architecture, and its role in object detection. In addition, this chapter also reviews the merits and demerits of ViTs models in object detection.

Chapter five5 introduces an evaluation of our case study, which is the DETR model. This chapter evaluates our DETR model for object detection, its architecture, and further discusses an experiment carried out on the COCO datasets. In addition, we review its comparison with other CNN-based models such as Faster-RCNN and RetinaNet.

Chapter six6 finally discusses the summary and conclusion of our overview on ViTs object detection. This chapter summarises the key findings and analyzes the results for future works in object detection and ViTs.

In all, this paper serves as a useful resource for future research, in the area of object detection using Visual transformers.

2 Related Work

Object detection has long been a study topic in computer vision, with classic algorithms relying heavily on convolutional neural networks (CNNs). This section includes an overview of relevant work in classic CNN-based object detection.

The Traditional CNN-Based Object Detection: CNNs have been the cornerstone of object detection for almost a decade and have experienced major breakthroughs. Some significant contributions in this domain include:

- **Region-based CNNs (R-CNN)**: To enhance object localization, the R-CNN family of models, including Fast R-CNN and Faster R-CNN, introduced the concept of region proposals. R-CNN and its variations, which exhibit enhanced capabilities and accuracy, are important turning points in the development of object detection[GDDM14]. But they also spurred greater investigation into quicker and more effective object detection frameworks, resulting in techniques like SSD (Single Shot Multibox Detector) and YOLO (You Only Look Once).
- **Single Shot MultiBox Detector (SSD)**:The Single Shot MultiBox Detector (SSD) is a cutting-edge object detection framework that transformed the industry by offering a quicker and more effective substitute for earlier region-based methods like R-CNN. SSD seeks to identify objects in images with different aspect ratios and sizes in a single network forward pass. SSD is a real-time object detection approach that integrates several feature maps of different scales to detect objects of varying sizes. SSD is utilized extensively in many different fields, including robotics and surveillance systems, due to its effectiveness in object detection, particularly in real-time applications. The way object detection frameworks have evolved has been greatly aided by their ability to strike a compromise between speed and accuracy. [LAE+16].
- **You Only Look Once (YOLO)**: The YOLO family of models, which included YOLOv3 and YOLOv4, offered a unified architecture for real-time object recognition by partitioning the picture into a grid and concurrently predicting bounding boxes and class probabilities. You Only Look Once (YOLO) is a revolutionary object detection method noted for its real-time capabilities and precise item location. YOLO offers a paradigm change in object detection, attempting to predict bounding boxes and class probabilities directly from an image in a single neural network run. YOLO's capacity to detect objects in real-time without sacrificing accuracy has made it useful in a variety of sectors, including surveillance, autonomous cars, and robots. Its continual growth with subsequent variants demonstrates continued attempts to increase speed and accuracy for practical applications.[RDGF16].
- **RetinaNet**: RetinaNet is a cutting-edge single-stage object detection architecture that is notable for its ability to handle a broad range of object sizes. It solves

earlier approaches' shortcomings, particularly in identifying tiny objects reliably while remaining efficient. To solve class imbalance difficulties, RetinaNet presented a one-stage object detection architecture with a focal loss function. The successful usage of FPN and the innovative Focal Loss function by RetinaNet solves issues experienced by previous single-stage detectors, resulting in outstanding performance in precisely detecting objects of varying sizes while preserving computing efficiency. This makes it an appealing option for a wide range of real-world object detection applications. [LGG+17].

While traditional CNN-based methods have shown amazing results in object detection, they frequently fail to capture long-range dependencies in data, which is where Vision Transformers come in.

3 Transformers

Transformers [VSP+17]revolutionized natural language processing with their debut, and their versatility also extends to computer vision applications like object detection. An extensive examination of transformers and their use in the context of object detection is given in this section. The structure is as follows: several transformer blocks with the same architecture make up the encoder and decoder, Encoders create input encodings, while decoders take all of the encodings and create the output sequence by including contextual information. Layer normalization, a feed-forward neural network, a shortcut connection, and a multi-head attention layer make up every transformer block. Each part of the transformer is explained in depth in the sections that follow.

3.1 Transformer Architecture

The transformer architecture was first presented for sequence-to-sequence tasks in natural language processing in the original paper by Vaswani et al. (2017). The self-attention mechanism, which enables the model to weigh various input sequence segments differently and successfully capture long-range dependencies, is the transformer's essential component [VSP+17].

Fig 3.1 shows the activity going on inside the Transformer architecture.

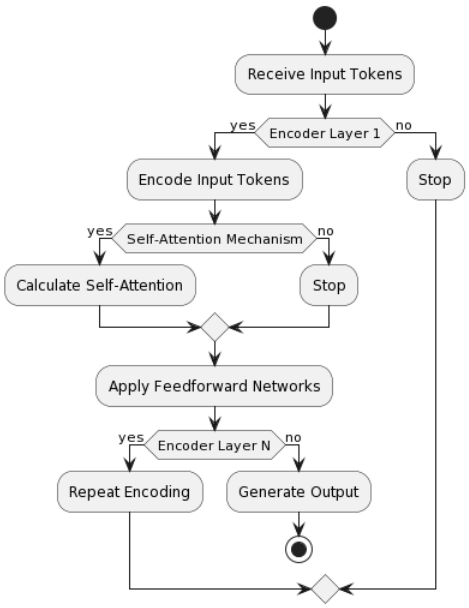


Figure 3.1: The Activity Diagram of the Transformer Architecture according to [VSP+17]

The transformer design has been modified to efficiently analyze image data in the context of object detection. In contrast to conventional convolutional methods, which depend on specific receptive fields, transformers work on the entire image at once. This perspective change from local analysis to comprehending the global context is especially helpful for tasks where relationships between items may extend over the whole image, such as object detection.

The transformer is a good choice for computer vision applications because of its innate ability to capture global dependencies, even though it was originally intended for sequential data. Vision Transformers (ViTs) are the result of this flexibility; they apply the transformer design directly to image data by partitioning it into fixed-size patches[DBK+21]. This break from grid-like convolutions creates new opportunities for object detection and presents an innovative method for contextual comprehension and feature extraction.

3.2 Positional Encodings in Transformer-based Image Processing

The intrinsic lack of sequential information in images is one of the fundamental difficulties in modifying transformers, which were initially intended for sequential data, for use in image processing applications. In order to overcome this constraint, Vaswani et al. (2017) introduced positional encodings, an essential element that provides transformer models with spatial information[VSP+17].

The use of positional encodings becomes crucial when employing Vision Transformers (ViTs) for object detection. In order to use the transformer architecture directly on picture patches, as suggested by Dosovitskiy et al. (2021), ViTs need a way to encode the spatial connections between these patches[DBK+21].

Positional encodings are included in ViTs input embeddings, giving the model critical information on the patch layout within the image. A transformer that relied just on patch locations would not be able to distinguish between distinct patches in the absence of such encodings. This is especially important for object detection, as reliable localization and object categorization depend on a clear grasp of spatial relations. The use of positional encodings is a crucial step in augmenting transformers' versatility for image-based operations. It enables ViTs to process images as a sequence of patches efficiently while preserving the spatial layout information of these patches.

3.3 Self-Attention Mechanism

The self-attention mechanism enables transformers to weigh the relevance of various items in the input sequence in relation to one another. Self-attention makes it possible for the model to comprehend the connections between various image areas when it comes to object detection. Understanding the global context is very useful for identifying things in complicated settings [VSP+17]. The model's attention mechanism enables it to focus on relevant regions of the input image, capturing both local and global dependencies.

Compared with traditional convolutional procedures, which depend on fixed receptive fields, this is different.

3.3.1 Self Attention

The self-attention here explains along the direction of natural language processing, this is basically to give us the foundation of how the Self Attention in the transformer operates. The input vector is initially converted into three distinct vectors in the self-attention layer: the key vector $\mathbf{k}$, the query vector $\mathbf{q}$, and the value vector $\mathbf{v}$ of dimension $d_q = d_k = d_v = d_{model} = 512$, Vectors generated from various inputs are then packed into three separate matrices, namely, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$. Following that, the attention function

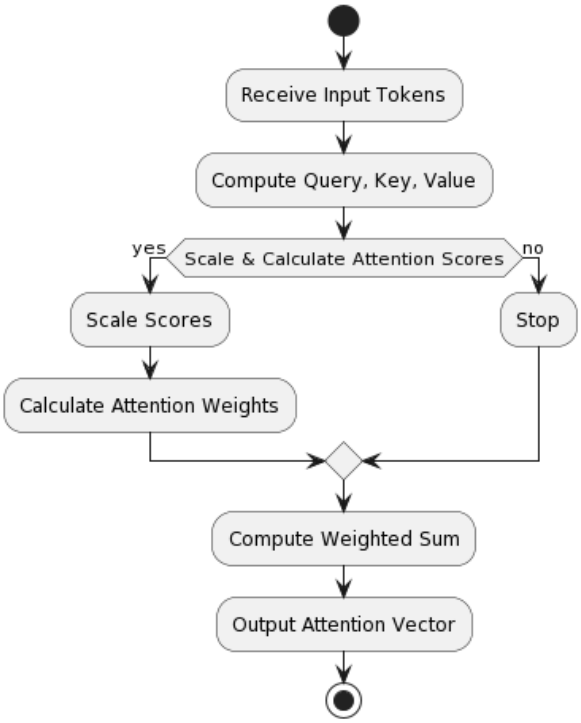


Figure 3.2: The Activity Diagram of the Self Attention according to[VSP+17]

between distinct input vectors is determined as follows and as illustrated in in the activity diagram in (Fig. 3.2). **NB**: This is an equation from [HWC+22] :

- Step 1: Compute scores between different input vectors with $\mathbf{S} = \mathbf{Q}\cdot\mathbf{K}^{\mathrm{T}}$;
- Step 2: Normalize the scores for the stability of gradient with $\mathbf{S}_n = \mathbf{S}/\sqrt{d_k}$;
- Step 3: Translate the scores into probabilities with softmax function $\mathbf{P} = \mathrm{softmax}(\mathbf{S}_n)$;
- Step 4: Obtain the weighted value matrix with $\mathbf{Z} = \mathbf{V}\cdot\mathbf{P}$.

The procedure may be combined into a single function

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\cdot\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\cdot\mathbf{V}. \tag{3.1}$$

The reasoning underlying Eq. (3.1) is straightforward. The following steps explain its processes:

- Step 1 computes scores between each pair of distinct vectors, and these values dictate how much attention we pay to other words while encoding the word in the present place.
- Step 2 normalizes the results to improve gradient stability and training.
- Step 3 converts the values to probabilities. Finally, the total of the probabilities is multiplied by each value vector. The following layers pay special attention to vectors with higher probability.

With the following exceptions, the encoder-decoder attention layer in the decoder module is comparable to the self-attention layer in the encoder module: The encoder module generates the key matrix K and the value matrix V, whereas the previous layer generates the query matrix Q.

It is worth noting that the previous procedure is insensitive to the location of each word, implying that the self-attention layer is incapable of capturing the positional information of words in a phrase. However, because sentences in a language are sequential, we must include positional information in our encoding. To solve this issue and acquire the word's final input vector, a positional encoding of dimension d_{model} is applied to the original input embedding.

The location is specifically represented using the equations below:

$$PE(pos, 2i) = sin\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right); \tag{3.2}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right), \tag{3.3}$$

where *pos* is the location of the word in a sentence and i is the current dimension of the positional encoding. In this method, each positional encoding element correlates to a sinusoid, allowing the transformer model to learn to attend by relative locations and extrapolate to larger sequence lengths during inference. Aside from the vanilla transformer's fixed positional encoding, learned positional encoding [GAG+17] and relative positional encoding [SUV18] are also used in other models [DCLT18], [DBK+21].

3.3.2 Multi-Head Attention

The multi-head attention mechanism is a critical component of transformer systems, allowing them to grasp subtle patterns and relationships within data. Multi-head attention, first presented by Vaswani et al. (2017) in the context of natural language processing, improves the expressive potential of transformers by allowing them to pay to multiple sections of the input sequence at the same time[VSP+17].

The multi-head attention mechanism is critical in comprehending the links between picture patches in the context of object recognition utilizing Vision Transformers (ViTs). Dosovitskiy et al. (2021) proved the efficacy of multi-head attention in ViTs, demonstrating its capacity to collect both local and global characteristics in an image[DBK+21].

3.3.3 Feed Forward Network(FFN)

The feed-forward network, a vital component of transformer architecture, is the second critical building element, following the self-attention mechanism. The feed-forward network, which Vaswani et al. (2017) first described in the transformer model, is critical in capturing complicated, non-linear interactions among encoded features[VSP+17]. The feed-forward network acts on each location of the input sequence separately, projecting the multi-dimensional feature representations learnt by the self-attention mechanism into a higher-dimensional space. This non-linear modification improves the model's ability to grasp complicated connections and learn hierarchical features.

4 Vision Transformers

Vision Transformers (ViTs) have received a lot of interest as a development of the transformer design for its use in computer vision applications, notably object detection. This section delves into Vision Transformers, their architecture, and their contributions to the field. Dosovitskiy et al. (2021) presented Vision Transformers as an alternative architecture for image recognition tasks[DBK+21]. Unlike standard convolutional algorithms, ViTs directly apply the transformer architecture to image patches, providing a novel viewpoint on feature extraction and global context comprehension. Fig. 4.1 shows the framework of ViT, in addition with the transformer encoder, which is illustrated in an activity diagram manner in Fig 4.2

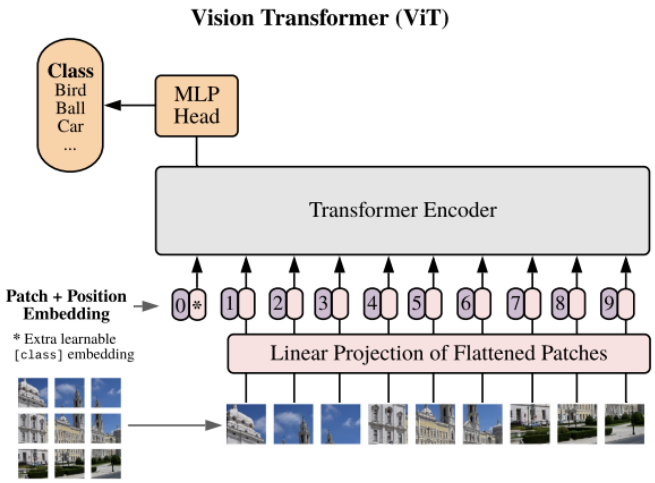


Figure 4.1: The Vision Transformer Framework [DBK+21]

The capacity of ViTs to treat images as sequences of fixed-size patches enables the model to capture both local and long-range dependencies within the image. It is worth noting that ViT simply employs the conventional transformer's encoder (except for the location for layer normalization), the output of which precedes an **MLP head**. Most of the time, ViT is pre-trained on huge datasets before being fine-tuned for downstream applications with less data.

When trained on mid-sized datasets like ImageNet, ViT produces moderate results, with accuracies that are a few percentage points lower than ResNets of comparable size. Transformers do not generalize well when trained on limited data because they lack some inductive biases present in CNNs, such as translation equivariance and localization. Researchers discovered, however, that training the models on large datasets (14 million to 300 million images) outperformed inductive bias. When pre-trained at a suitable size.

ViT, for example, reached or even exceeded state-of-the-art performance on numerous image recognition benchmarks when pre-trained on the JFT-300M dataset. It achieved an accuracy of 88.36% on ImageNet and 77.16% on the VTAB suit of 19 tests.

4.1 Patch Embeddings and Positional Encodings in ViTs

The method of transforming images into sequences of fixed-size patches in Vision Transformers (ViTs) is a significant divergence from traditional convolutional approaches[DBK+21]. This transformation allows ViTs to evaluate images in a way similar to natural language processing, with each patch acting as a "word" in the visual context. The linear embedding of these patches serves as the transformer's input sequence, establishing the framework for later self-attention procedures. This **patch-based** strategy was presented by Dosovitskiy et al. (2021), who demonstrated its usefulness in ViTs for image recognition[DBK+21]. ViTs may capture both local and global aspects by breaking images down into digestible patches, encouraging a full comprehension of visual material.

Positional encodings, proposed by Vaswani et al.(2017) to supplement this patch-based representation[VSP+17]. In the absence of intrinsic sequential information, positional encodings provide ViTs with critical spatial information regarding the layout of patches within the images. This innovation allows ViTs to keep track of the image's structure, which is critical for tasks like object detection, where the spatial connection between items is critical.

The inclusion of positional encodings into input embeddings is critical in improving ViTs' spatial awareness. Positional encodings help the model grasp the relative location of items in an image by encoding the spatial coordinates of each patch.

This patch-based paradigm, in conjunction with positional encodings, demonstrates ViTs' adaptability in dealing with various visual inputs and extracting important characteristics from images. Ongoing research focuses on improving these methods in order to increase ViTs' adaptation to different datasets and scene complexity.

4.2 Tranformer encoder in ViTs

The Transformer encoder is the key module of Vision Transformer (ViT) architectures, responsible for deriving meaningful representations from input images using self-attention processes and position-wise feedforward networks. The Transformer encoder is the primary building piece in charge of tokenizing input images and allowing the model to interpret and extract meaningful representations from them. This module modifies the Transformer architecture, which was originally built for natural language processing problems, to analyze imagine input using self-attention mechanisms and position-wise feedforward networks[DBK+21]. The Transformer encoder enables ViTs to process tokenized images, in which the image is segmented into fixed-size patches, linearly embedded, and supplied into the Transformer layers. The self-attention mechanism is critical for acquiring global information, whereas feedforward networks aid in the refinement and processing of these learned representations at each point. The Transformer encoder is a key component in achieving state-of-the-art performance in various computer vision tasks, especially

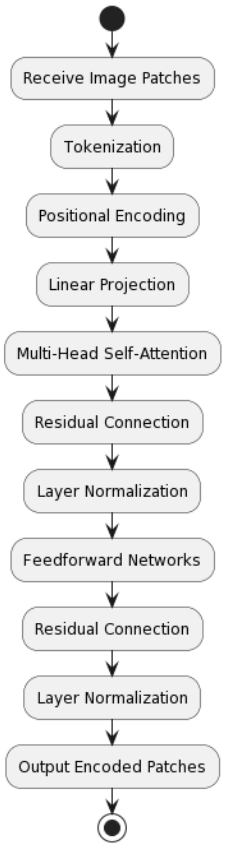


Figure 4.2: Activity diagram for Vision Transformer Encoder according to[DBK+21]

object detection, due to its adaptability to different tokenized input modalities, ability to model relationships between tokens efficiently, and scalability across diverse datasets. The Transformer encoder is made up of several layers of self-attention blocks, each with self-attention and feedforward sublayers. fig 4.2 relates the activity going on inside the Transformer encoder in ViT.

4.2.1 Self-Attention Mechanisms in ViTs

The use of self-attention mechanisms, a crucial component inherited from the transformer design, is critical to the success of vision transformers (ViTs). ViTs may use this method to assign variable degrees of importance to distinct patches within an image, making it easier to collect both local and global contextual information. The self-attention mechanism, first proposed by Vaswani et al. (2017) in the context of natural language processing, allows ViTs to assess the relevance of distinct patches in relation to each other[VSP+17]. This ability aids in grasping the complicated links between visual aspects, allowing ViTs to recognize complex patterns and structures in images.

Dosovitskiy et al. (2021) proved the efficacy of self-attention in ViTs for image recognition, demonstrating its capacity to capture long-range dependencies [DBK+21]. This correlates to the model's ability to recognize relationships between items scattered over an image in the domain of object recognition, contributing to more accurate and context-aware predictions.

The self-attention mechanism enables ViTs to focus on important segments of the input image, promoting global context awareness. This deviation from fixed receptive fields in standard convolutional processes is especially useful in instances when objects have varying scales and spatial distributions.

In conclusion, the incorporation of self-attention processes in ViTs is critical for their object recognition ability, allowing the model to examine images holistically and generate informed predictions based on both local and global contextual information.

We will also briefly give details on the work of the **MLP** along the path of the transformer encoder after the patch and positional encodings have processed their inputs.

4.2.2 The MLP (Multi-Layer Perceptron)

Within the Transformer's encoding step, the MLP (Multi-Layer Perceptron) layer is a vital component responsible for digesting tokenized inputs and improving the model's capacity to grasp nuanced patterns and complex connections. [DBK+21], [TCD+21]. The MLP layer, also known as the position-wise feedforward network in the Transformer design, functions after the self-attention mechanism within each Transformer encoder layer. This MLP layer is made up of numerous fully connected neural network layers as well as non-linear activation functions as GELU (Gaussian Error Linear Unit) or ReLU (Rectified Linear Unit) [DBK+21]. The MLP layer's principal work is to handle the output of the self-attention mechanism by applying position-wise modifications to tokenized representations. The MLP executes pointwise operations over its fully linked layers, improving the network's capacity to represent local patterns and interactions between tokens [DBK+21],[TCD+21].

The Transformer design makes use of the MLP layer's ability to incorporate non-linearities and process tokenized embeddings by incorporating these position-wise feedforward networks, allowing the model to learn more expressive and contextually rich representations from the input data.

Also worthy of note is the **MLP head**

The MLP head: also known as the "MLP head after the self-attention mechanism," is a critical component that is in charge of processing tokenized image representations prior to final classification or regression. This segment follows the self-attention layers and includes many completely linked levels. These layers contribute to the network's capacity to recognize complicated visual patterns and extract higher-level abstractions from visual input by processing tokenized embeddings obtained from picture patches. Non-linear activation functions, such as GELU (Gaussian Error Linear Unit) or ReLU (Rectified Linear Unit), are often applied after each fully connected layer, adding non-linearity and increasing the expressive capacity of the network.[TCD+21]

Within ViTs, the MLP head is critical in global feature processing and learning task-specific representations from tokenized embeddings. The architectural design and depth of the ViT have a substantial impact on its capacity to capture complex visual patterns and generalize across varied datasets[CMS+20].

4.2.3 The multi-head attention mechanism in ViTs

The multi-head attention mechanism is critical in capturing complicated links and dependencies between tokens inside input sequences, allowing the model to recognize global context and derive meaningful representations [DBK+21],[CMS+20].

The multi-head attention mechanism in ViTs is an important component of the Transformer encoder, which is made up of many self-attention heads working in simultaneously. Each self-attention head learns discrete associations between tokens by linearly projecting the input embeddings into query, key, and value vectors [DBK+21].

These numerous attention heads enable the ViT to pay to different regions of the input sequence at the same time, allowing the model to capture distinct characteristics and patterns at different levels of abstraction. The multi-head attention mechanism improves the model's ability to grasp subtle spatial linkages and long-term dependencies within the input sequence by processing tokens in parallel and aggregating information from diverse perspectives [DBK+21],[TCD+21].

This parallel processing and aggregation of information across many attention heads greatly contributes to the ViT's capacity to extract rich and context-aware representations from input data, allowing for strong performance across a variety of computer vision applications.

4.3 Merits/Advantages of ViTs in Object Detection

ViTs bring several advantages to the realm of object detection, such as:

- **Global Context Understanding**: ViTs' self-attention mechanisms allow them to comprehend the links between items in a scene, taking into account the global context for more accurate detection[DBK+21].
- **Scalability**: ViTs have shown scalability in handling large-scale datasets and complicated visual tasks, demonstrating their promise for real-world applications [DBK+21].
- **Adaptability to Varying Resolutions**: ViTs can adapt to images of varied resolutions, making them useful for object detection tasks using a wide range of input data[DBK+21].

[CMS+20].

4.4 Challenges/Demerits and future direction of ViTs in object Detection

While Vision Transformers (ViTs) have shown amazing progress in object detection and image recognition, a number of challenges remain, limiting their mainstream use and future development. These challenges are as follow:

- **Computational Intensity and Efficiency**: The computational intensity of ViTs during both training and inference is a primary challenge. ViTs are resource-intensive due to the quadratic complexity of self-attention mechanisms, which frequently need substantial computational power and memory capacity[DBK+21]. This presents difficulties for real-time applications and devices with limited resources. Exploring more efficient transformer topologies, improving training methodologies, and designing hardware solutions specialized to ViTs are all part of addressing this problem.
- **Scalability to Large Datasets**: ViTs have demonstrated scalability to large-scale datasets, however sustaining performance across different and broad datasets[DBK+21] presents issues. Adapting ViTs to deal with datasets with various object sizes, resolutions, and complexity is still a work in progress. Future research should concentrate on increasing ViTs' generalization capabilities, ensuring robust performance over a broad range of real-world scenarios.
- **Limited Spatial Hierarchies**: Because of their layer-wise nature, traditional convolutional architectures automatically record hierarchical spatial features. ViTs, on the other hand, process images in a patch-based fashion, which may restrict their capacity to capture spatial hierarchies efficiently[DBK+21]. Exploring innovative attention mechanisms or hybrid architectures that may explicitly describe hierarchical structures inside images, boosting ViTs' comprehension of spatial linkages, is one approach to addressing this difficulty.
- **Interpretability and Explainability**: The intricate structure of self-attention mechanisms in ViTs might pose interpretability and explainability challenges[DBK+21]. Understanding how ViTs make decisions and offering meaningful insights into their internal representations is vital for establishing confidence in real-world applications, particularly in areas where interpretability is required. Future research should concentrate on establishing ways for interpreting and explaining ViTs decisions in order to make them more transparent and understandable.
- **Robustness to Noisy and Adversarial Inputs**: Like other deep learning models, ViTs may be vulnerable to noisy or adversarial inputs[DBK+21]. A significant difficulty is ensuring the robustness of ViTs in the event of variances, distortions, or adversarial attacks. Developing robust training procedures, including data augmentation techniques, and investigating adversarial training methods are all ways to solve this issue and improve the reliability of ViT-based object detection systems.
- **Transfer Learning Across Diverse Domains**: While transfer learning has been effective in the context of ViTs, there are still challenges in transferring knowledge across extremely different fields [DBK+21]. Addressing domain gaps and fine-tuning procedures are required when adapting ViTs for specific domains such as medical imaging or satellite imagery. Overcoming these challenges will allow ViTs to be used in more areas than just normal object detection datasets.
- **Large-Scale Annotation Requirements**: For efficient training, ViTs frequently require large-scale annotated datasets[DBK+21]. Acquiring labelled data at scale can be time-consuming and resource-intensive, making ViTs inaccessible to academics and practitioners with minimal resources. Developing strategies for more effective

data consumption, such as semi-supervised learning or unsupervised pre-training, can help to overcome this challenge and increase the application of ViTs in a variety of contexts.

- **Future Directions**: Despite these challenges, Vision Transformers continues to stimulate research and development. Future prospects include collaborative efforts to address the aforementioned challenges, refine ViT architecture, and investigate multidisciplinary applications, with the goal of ultimately increasing the capabilities of ViTs in the developing environment of computer vision.

5 Evaluation

This chapter covers the review of our use case, DETR Model, which is an object detection method on Transformer-based set prediction for Detection.

5.1 DETR Architecture

The detection Transformer (DETR) architecture is an innovative application of transformer-based models to object detection tasks. The detection transformer (DETR) suggested by Carion et al[CMS+20].redesigns the framework of object detection as a pioneer for transformer-based detection methodologies. DETR, a simple and fully end-to-end object detector, approaches the challenge of object detection as an intuitive set prediction problem, omitting standard hand-crafted components such as anchor generation and non-maximum suppression (NMS) post-processing. Let's look at its important architectural elements as shown in fig 5.1:

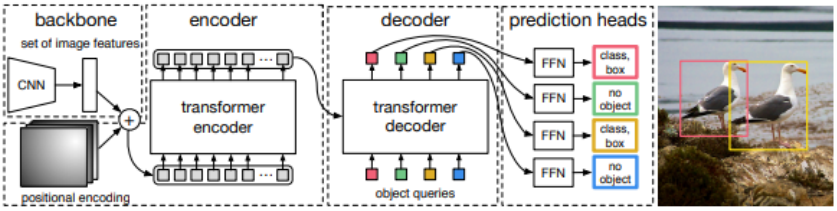


Figure 5.1: The DETR architecture [CMS+20]

- **Backbone CNN Encoder**: In most cases, DETR begins with a Convolutional Neural Network (CNN) serving as an encoder, extracting features from the input image. This backbone CNN, which is comparable to ResNet or other architectures, analyzes the image and outputs a feature map that represents the visual content.
- **Transformer Encoder-Decoder Architecture**: DETR makes use of a transformer-based architecture with an encoder-decoder configuration. The encoder processes the visual information extracted from the image by the CNN backbone, while the decoder predicts object bounding boxes and class labels.
- **Self-Attention Mechanism**: The self-attention mechanism is key to the transformer architecture. DETR may record global contextual links between image characteristics and object searches using this method. DETR can reason holistically about object location and classification by attention to multiple regions of the image at the same time.

- **Object Queries**: DETR uses learnable object queries instead of conventional anchor boxes. These queries, which are analogous to tokens in language models, describe possible locations and properties of objects inside the image. The model responds to these queries in order to identify and categorize objects, removing the requirement for handmade priors.
- **Positional Encodings**: DETR, like other transformer-based models, utilizes positional encodings to offer spatial information regarding feature layout. This is critical for preserving object spatial connections in the absence of explicit spatial information inherent in sequential data.
- **Simple feed-forward networks (FFNs)**: The final predictions are computed using **FFNs**, which include the bounding box coordinates and class labels to indicate the exact type of object (or that no object exists). DETR decodes N items in parallel, as opposed to the original transformer, which computes predictions sequentially.
- **Bipartite Matching Loss**: DETR proposes a bipartite matching loss function for comparing expected outputs to ground-truth annotations. This loss promotes end-to-end training, allowing the model to learn how to best assign items to predicted bounding boxes.

DETR outperforms the popular and well-established Faster R-CNN [RHGS15] baseline on the COCO benchmark in terms of object detection, giving comparable accuracy and speed.

The next review will be on the experiment carried out using COCO datasets with DETR, and its comparison with other models such as Faster-RCNN and Retina.net.

5.2 Experiment

In a quantitative assessment on COCO, it was demonstrated that DETR outperforms Faster R-CNN [RHGS15] and RetinaNet [LGG+17]. Then, using insights and qualitative results, a thorough ablation analysis of the architecture and loss was presented.

DATASET: The experiment was performed on COCO 2017 detection dataset, which contains 118k training images and 5k validation images. Where each image is annotated with bounding boxes. On average, there are 7 instances per image, with up to 63 instances in a single image in the training set, varying in size from small to large on the same images. If no criteria are given, AP is reported as bbox AP, which is the integral measure over several thresholds. Also validation AP was reported at the latest training epoch for comparison with other models, and in ablations, the median was reported over the last 10 epochs.

TECHNICAL DETAILS FOR THE ASSESSMENT: DETR was trained with AdamW [LH17], thereby, setting the initial transformer's learning rate to 10^{-4}, the backbone's learning rate to 10^{-5}, and the weight decay rate to 10^{-4}. The backbone is an ImageNet-pre-trained ResNet model[HZRS16] from torchvision with frozen batchnorm layers, and all transformer weights are initialized with Xavier init [GB10]. The results was reported with two different backbones: a ResNet50 and a ResNet-101. The equivalent models are called DETR and DETR-R101 respectively. In addition, the feature resolution

was improve by eliminating a stride from the first convolution of the last stage of the backbone and adding a dilation to it, as per [LQD+17]. The equivalent models are called DETR-DC5 and DETR-DC5-R101 (dilated C5 stage) respectively. This modification boosts the resolution by a factor of two, boosting performance for small objects at the expense of a 16x increase in the encoder's self-attentions, resulting in an overall 2x increase in computational cost. Fig 5.2 compares the FLOPs of these models, Faster R-CNN and RetinaNet, in detail. The input images were resized using scale augmentation, with the shortest side being at least 480 pixels and the longest being 1333 pixels [WKM+19]. random crop augmentations Were also used during training to enable the encoder to learn global connections through self-attention, enhancing performance by about 1 AP. A train image is specifically cropped with probability 0.5 to a random rectangular patch, which is subsequently scaled to 800-1333. The transformer is trained with a dropout of 0.1 as the default. Some slots indicate an empty class during inference. To optimize for AP, the prediction of these slots with the second-highest scoring class was overridden, and the matching confidence was used. When compared to filtering out empty spaces, this enhances AP by 2 points. For the ablation studies, a 300-epoch training schedule was adopted, with a learning rate reduction by a factor of 10 after 200 epochs, where every epoch is a single pass over all training images. Training the baseline model for 300 epochs takes 3 days on 16 V100 GPUs, with 4 photos per GPU (for a total batch size of 64). It was trained for 500 epochs with a learning rate decline after 400 epochs for the longer schedule used to compare with Faster R-CNN, which increases AP by 1.5 points.

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	205/18	38M	38.7	58.0	41.5	23.3	42.3	50.3
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
RetinaNet+	205/18	38M	41.1	60.4	43.7	25.6	44.8	53.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Figure 5.2: On the COCO validation set, RetinaNet and Faster R-CNN with ResNet-50 and ResNet101 backbones were compared. The results for models in Detectron2 [WKM+19] are shown in the top area, whereas the results for models using GIoU [RTG+19], random crops train-time augmentation, and the extended 9x training schedule are shown in the center section. DETR models outperform extensively tweaked Faster R-CNN baselines in terms of APS but considerably improved APL. Torchscript models are used to calculate FLOPS and FPS. Without R101 in the name, the results equate to ResNet-50. [CMS+20]

5.3 DETR Comparison with Faster R-CNN and RetinaNet

Transformers are often trained using Adam or Adagrad optimizers with extremely lengthy training schedules and dropout, and DETR is no exception. Faster R-CNN, on the other hand, is trained with SGD with limited data augmentation, and we are not aware of any successful Adam or dropout applications. Regardless of these distinctions, we strive to strengthen our foundations. To make it more like DETR, we add generalized IoU[RTG+19] to the box loss, as well as the same random crop augmentation and extended training that has been shown to boost outcomes [HGD19]. Figure 5.2 depicts the results. In the top part, we exhibit Detectron2 Model Zoo [WKM+19] results for models trained with the 3x schedule. We provide results (with a "+") for the identical models trained with the 9x schedule (109 epochs) and the mentioned improvements, which adds 1-2 AP in total. The findings for numerous DETR models are shown in the final portion of fig 5.2. We chose a model with 6 transformer and 6 decoder layers of width 256 and 8 attention heads to be equivalent in terms of parameter count. This model, like Faster R-CNN with FPN, contains 41.3M parameters, 23.5M of which are in ResNet-50 and 17.8M in the transformer. Despite the fact that both Faster R-CNN and DETR are expected to improve more with additional training, we may infer that DETR can compete with Faster R-CNN with the same amount of parameters, reaching 42 AP on the COCO val subset. DETR achieves this by increasing AP_L (+7.8), however note that the model is still underperforming in AP_S (-5.5). DETR-DC5, with the same number of parameters and FLOP count, has greater AP but falls severely short in AP_S. The results on the ResNet-101 backbone are also comparable.

5.4 Ablation Studies

The transformer decoder's attention mechanisms are the fundamental components that explain relationships between feature representations of distinct detections. In our ablation study, we investigate how various aspects of our design and loss affect final performance. We used a ResNet-50-based DETR model with six encoder and six decoder layers and a width of 256 for the investigation. The model has 41.3M parameters, achieves 40.6 and 42.0 AP on short and long schedules, and operates at 28 FPS, which is comparable to Faster R-CNN-FPN with the same backbone.

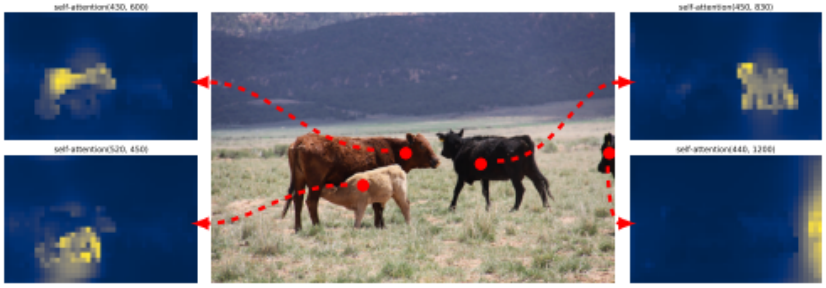


Figure 5.3: Encoder self-attention for a collection of reference locations. The encoder can distinguish between individual occurrences. Prediction done using baseline DETR on a validation image. [CMS+20]

Number of encoder layers: By varying the number of encoder layers, we assess the relevance of global image-level self-attention. Without encoder layers, total AP falls by 3.9 points, with a larger reduction of 6.0 points for big objects. We think that the encoder is critical for object disentanglement because it uses global scene reasoning. Figure 5.3 depicts the attention maps of a trained model's final encoder layer, concentrating on a few locations in the picture. The encoder seems to already segregate instances, which facilitates object extraction and location for the decoder.

Number of decoder layers: As a result of applying auxiliary losses after each decoding layer, the prediction FFNs are designed to anticipate objects based on the outputs of each decoder layer. We evaluate the items that would be predicted at each stage of decoding to determine the relevance of each decoder layer (Fig. 5.4). After each layer, both AP and AP50 improve, resulting in a very large +8.2/9.5 AP improvement between the first and last layer. DETR does not require NMS because of its set-based loss. To test this, we conduct a normal NMS method with default parameters [WKM+19] for the decoder outputs. NMS enhances prediction performance from the first decoder. This is due to the fact that the transformer's single decoding layer is unable to compute any cross-correlations between the output parts, rendering it prone to producing several predictions for the same item. The self-attention mechanism over the activations allows the model to suppress duplicate predictions in the second and subsequent layers. As depth grows, we see that the benefit of NMS declines. It harms AP in the last layers by removing real positive predictions wrongly.

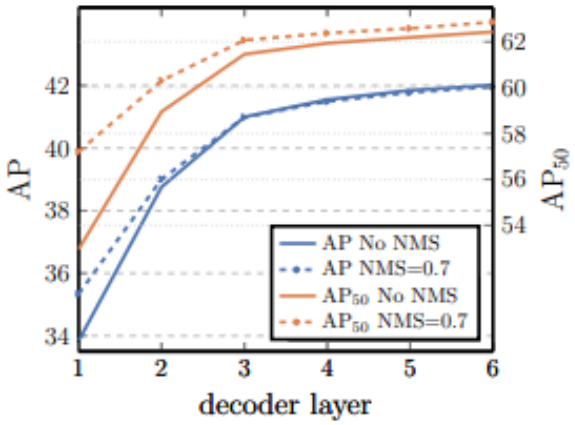


Figure 5.4: In a long schedule baseline model, AP and AP50 performance after each decoder layer. This figure validates the fact that DETR does not require NMS by design. NMS reduces AP in the last layers by deleting TP predictions, but enhances it in the initial layers, where DETR cannot eliminate double predictions.[CMS+20]

Fig.5.6 depicts decoder attention in the same way as encoder attention is depicted, by coloring attention maps for each anticipated item in various colors. We find that decoder attention is quite local, focusing on object extremities such as heads or legs. We hypothesize that once the encoder has split instances using global attention, the decoder only has to pay attention to the extremities in order to extract the class and object boundaries.

Importance of FFN: FFN within transformers may be seen as 1X1 convolutional layers,

comparable to attention-enhanced convolutional networks [BZV+19]. We try to eliminate it completely, leaving just the transformer layers in focus. We conclude that FFN are vital for attaining good outcomes by lowering the number of network parameters from 41.3M to 28.7M, leaving just 10.8M in the transformer.

Importance of positional encodings: In our model, there are two types of positional encodings: spatial positional encodings and output positional encodings (object queries). We test various combinations of fixed and learnt encodings, with the results shown in the appendix. Because output positional encodings are essential and cannot be eliminated, we experiment with passing them only once at decoder input or adding them to queries at each decoder attention tier. In the first trial, we totally eliminated spatial positional encodings and passed output positional encodings at input, and the model still achieves more than 32 AP while losing 7.8 AP compared to the baseline. Then, as in the original transformer[VSP+17], we send fixed sine spatial positional encodings and output encodings at input once and show that this results in a 1.4 AP loss when compared to passing the positional encodings directly in attention. Similar results are obtained when learned spatial encodings are transferred to the attentions. unexpectedly we discover that not passing any spatial encodings in the encoder results in a modest AP loss of 1.3 AP. When we provide the encodings to the attentions, they are shared across all layers, and the output encodings (object queries) are always learnt. Given these findings, we infer that transformer components such as global self-attention in encoders, FFN, numerous decoder layers, and positional encodings all contribute considerably to final object identification performance.

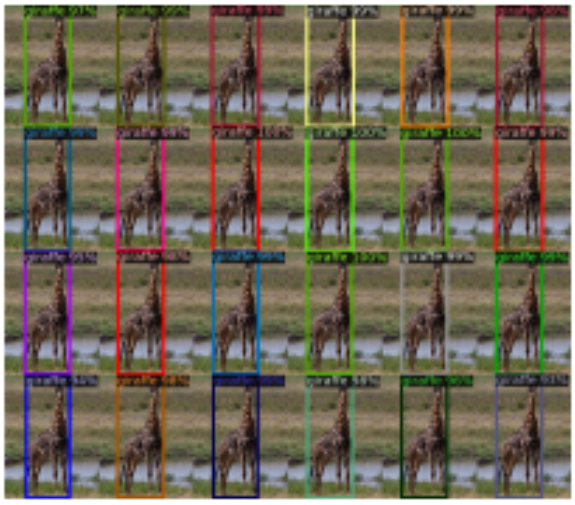


Figure 5.5: Generalization outside of distribution for unusual classes. Despite the fact that no image in the training set has more than 13 giraffes, DETR has no trouble generalizing to 24 or more instances.[CMS+20]

Generalization to unseen numbers of instances: Some COCO classes are poorly portrayed by having many instances of the same class in the same image. In the training set, for example, there are no images with more than 13 giraffes. To test DETR's generalization capabilities, we generate a synthetic image4 (i.e Figure 5.5). Our model is capable of detecting all 24 giraffes in the image, despite the fact that they are clearly

out of distribution. This experiment indicates that each object query lacks substantial class-specialization.

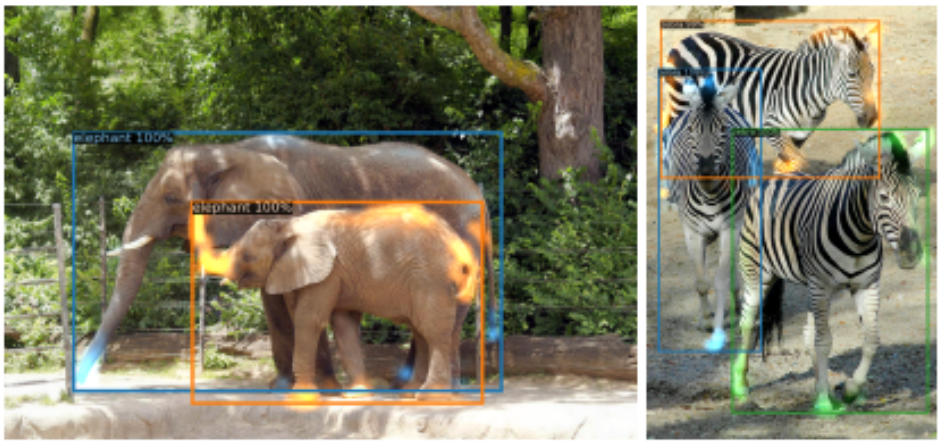


Figure 5.6: Decoder attention visualization for each predicted object. The DETR-DC5 model is used to make predictions. Decoders often focus on object extremities like legs and heads..[COCO dataset]

5.5 A running implementation of the model

The following figures are the results of the implementation of the model, showing the DETR detection, encoder self-attention mechanism weights and encoder self-attention itself.

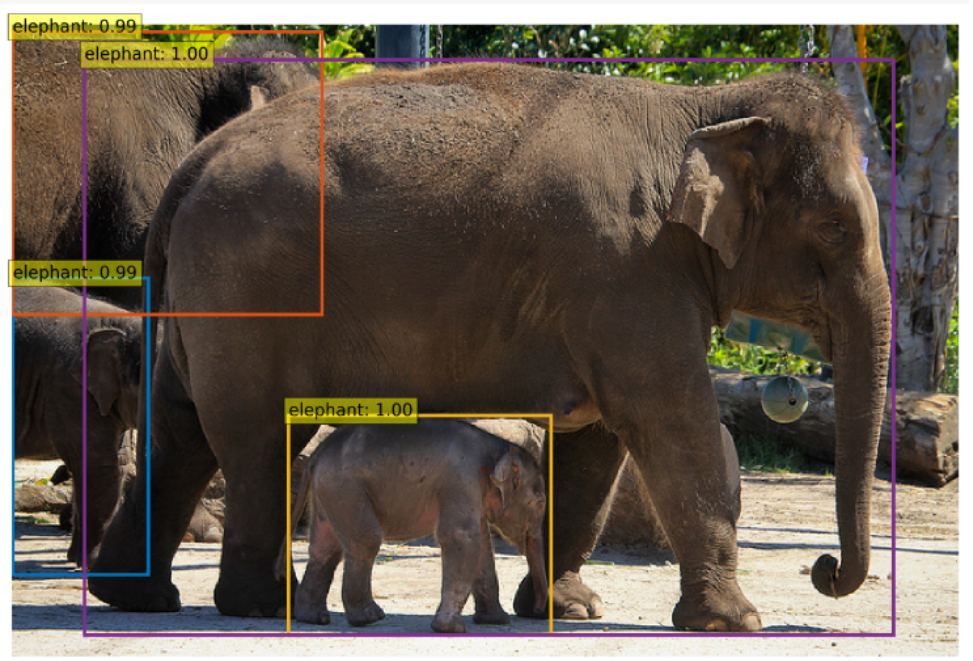


Figure 5.7: DETR Detection[COCO dataset]

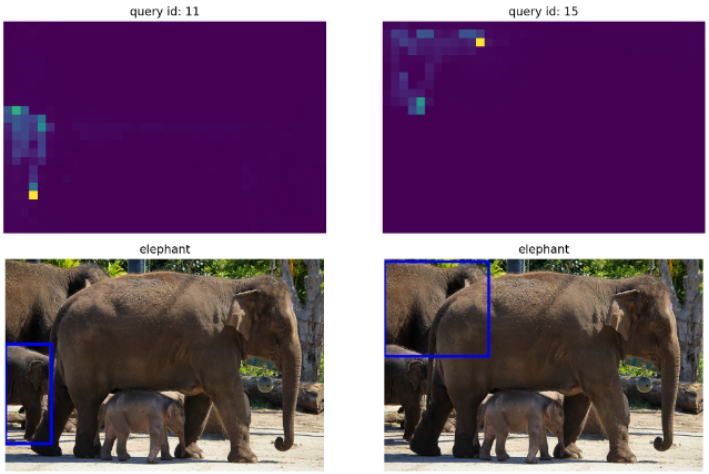
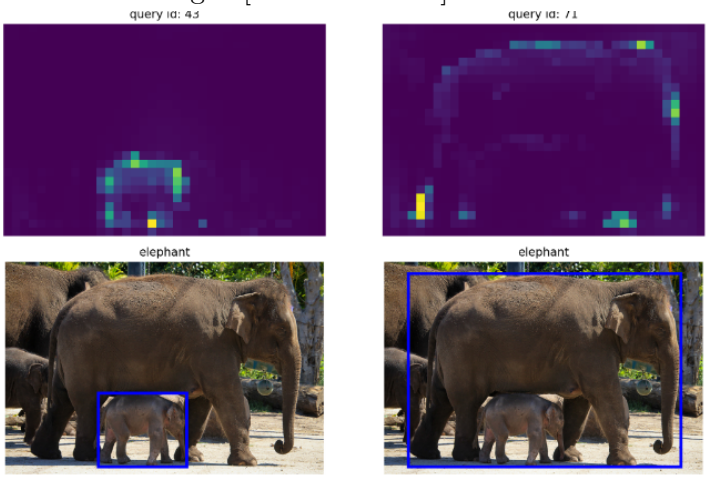


Figure 5.8: encoder self-attention mechanism weights[COCO dataset]



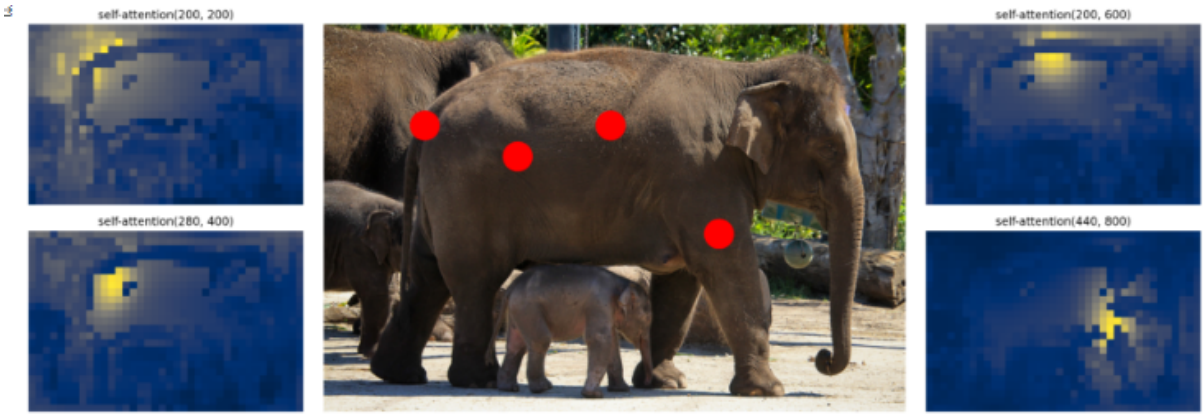


Figure 5.9: encoder self-attention[COCO dataset]

6 Outlook and Summary

In this project, we have discussed Object detection using vision transformers extensively. In the first chapter, we thoroughly introduced vision transformers, its objectives and goals. We then went further to introduce in the second chapter, the related works, how the traditional-based CNN model has been the backbone for object detection and how transformers-based has revolutionised the industry in the past decades.

The third chapter saw the evaluation of Transformers, its architecture, and methods. We dived deeply into explanations, to give us the basic structure we need to build on our proposed vision transformers with object detection.

We then proceeded with the main topic of this project in chapter four, the vision transformers. We extensively evaluated its definition, architecture, and methods. We further discussed its pros and cons and possibly future directions.

Chapter five welcomed the evaluation of our chosen case study, The DETR Model which we compared with traditional CNN-Based Faster-RCNN and Retina on COCO dataset.

DETR, a new architecture for object detection systems based on transformers and bipartite matching loss for direct set prediction, was presented. On the difficult COCO dataset, the technique delivers equivalent results to an improved Faster R-CNN baseline. DETR is simple to deploy and has a flexible design that produces competitive outcomes. Furthermore, it performs substantially better on large objects, most likely due to the processing of global information conducted by self-attention.

In conclusion, this new design for detectors also comes with new challenges, in particular regarding training, optimization and performances on small objects. Current detectors require several years of improvements to cope with similar issues, and we expect future work to successfully address them for DETR.

7 Appendix

Find The github repository here

7.1 Code in detail

`https://github.com/DrLeozeez/ObjectDetection_DETR/tree/main`

Bibliography

[ADH+21] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[ATC+21] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

[BMR+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[BZV+19] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.

[CMS+20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[CZJ+22] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2021.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[FXM+21] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers.

In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[GAG+17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[HGD19] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019.

[HWC+22] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[HZ21] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pages 4487–4499. PMLR, 2021.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[LGG+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[LH17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[LLG+19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[LQD+17] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[RTG+19] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[SUV18] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[TCD+21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WKM+19] Y Wu, A Kirillov, F Massa, WY Lo, and R Girshick. Detectron2 [www document]. *URL https://github. com/facebookresearch/detectron2 (accessed 3.3. 21)*, 2019.

[ZWZ19] Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*, 2019.

List of Figures

Affidavit

I Abdul-Azeez Olanlokun herewith declare that I have composed the present paper and work by myself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form has not been submitted to any examination body and has not been published. This paper was not yet, even in part, used in another examination or as a course performance. .
Lippstadt, December 3, 2023

Abdul-Azeez Olanlokun