Towards Improving Object Detection with End-to-End Transformers

Rituraj Singh
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
riturajs@andrew.cmu.edu

Tushar Vatsa
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
tvatsa@andrew.cmu.edu

Vivek Aswal
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
vaswal@andrew.cmu.edu

Wallace Dalmet
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
wdalmet@andrew.cmu.edu

Abstract

Convolutional neural networks have been proven to be extremely effective for end-to-end object detection tasks. Lately, the onset of transformers has opened new dimensions in improving the performance of the existing object detection networks. Although the application of transformers has been in natural language processing predominantly, recent research works have suggested that they are capable of achieving comparable, if not better, performance on computer vision problems while requiring substantially smaller number of parameters. In this project, we will develop pure transformer networks by replacing traditional CNN backbone architecture with a visual transformer classifier in DETR for object detection and measure their effectiveness on popular large-scale datasets.

1 Introduction

Object detection is a perennial task in computer vision. From hand crafted features to convolutional neural networks, research in solving this task has accomplished major milestones. In this project, our goal is to implement a novel architecture for object detection consisting of only transformers and measure its performance with recent transformer based object detectors and competitive baseline models having different variants of transformers.

Transformers have [1] led major breakthroughs in natural language processing. However, the application of pure transformers in computer vision problems is a direction that has largely been untapped. Unlike recurrent neural networks, transformers have the liberty to parallelize the inputs, thereby overcoming the constraint of sequential inputs. Object detection architectures that are comparable to the state-of-the-art are dependent on various hand-engineered features such as non-maximum suppression processing and anchor generation. Building end-to-end object detectors is a challenging task. However, the DETR (DEtection TRansformer) [2] architecture recently eliminated the need for explicit feature engineering through the application of CNNs coupled with a Transformer thereby converting the task of object detection into a direct set prediction problem. In 2020, Deformable DETR [3], a variant of the original DETR achieved competitive performance when pitched against models that rely heavily on convolutional layers, heuristics and feature engineering.

As of March 2021, the Swin Transformer [4] has achieved a staggering box AP score of 58.7 on the COCO dataset which showcases the remarkable effectiveness of the application of transformers in object detection.

The goal of this project is to demonstrate that the need for CNNs can be eliminated and a pure transformer can be used for object detection using novel modifications in the DETR architecture.

2 Literature Review

Previous research of object detection can be broadly grouped into two categories, namely, two-stage and one stage detectors. RCNN [5], Fast RCNN [6], Faster RCNN [7] and its variants [8, 9, 10] are two stage detectors which adopts a Region Proposal Network (RPN) to generate region proposals followed by proposal prediction. YOLO [11], SSD [12] are one stage detectors which remove the proposal generation stage and performs object classification and location estimation directly in images. These methods rely on many heuristics and are suffered by imbalanced loss [13], hand-crafted label assignment and complex non-maximum suppression (NMS) [14] post processing to remove redundant features during inference.

Different from the above methods, DEtection TRansformer (DETR) is fully end-to-end pipeline and a set-based global loss framework that forces unique predictions via bipartite matching, and a transformer encoder-decoder architecture to generate set predictions [15] directly from the image without any hand-crafted components. DETR uses a CNN backbone (Resnet) [16] to extract compact feature representation from images which is fed into an encoder-decoder transformer followed by a feed forward network that makes the final detection prediction. However, object detection using a Detection Transformer (DETR) suffers from slow convergence when compared to one-stage and two-stage object detectors. Modifications to DETR architecture, namely, Deformable DETR only attend to a small set of key sampling points around a reference thereby improving the convergence speed of DETR. TSP [17] investigate the causes of the optimization difficulty in the training of DETR and attributes the issue of slow convergence to Hungarian loss and the Transformer cross attention mechanism and attempt to overcome them by combining RCNN- or FCOS-based methods with DETR, the resulting TSP-FCOS and TSP-RCNN architectures achieves fast convergence and better performance. Other variations to DETR for reducing computation cost for high resolution inputs are ACT [18] which cluster the query features adaptively using Locality Sensitive Hashing (LSH) and approximate the query-key interaction using the prototype-key interaction, SMCA [19] conducts regression aware co-attention in DETR by constraining co-attention responses to be high near initially estimated bounding box locations.

ViT [20] is a transformer architecture that attempts to replace the CNN backbone for feature extraction by applying a standard transformer directly to images by splitting the image into patches and providing the sequence of linear embeddings of these patches as input to transformer for image classification task. ViT attains comparable result to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train. However, the good performance of ViT require training on large-scale datasets and its complexity increases quadratically with the image size thereby limiting its use as a backbone network. DeiT [21] improves on ViT by introducing a teacher-student strategy which relies on a distillation token ensuring that the student learns from the teacher through attention and achieves good results on smaller datasets. Swin Transformer, a hierarchical architecture transformer computes representation with shifted windows and has linear computational complexity with respect to image size. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection that enables Swin Transformer to serve as a general-purpose backbone for computer vision.

3 Datasets

In this project, the chosen datasets are COCO and CIFAR. The Common Objects in COntext (COCO) dataset contains 91 common object categories with 82 of them having more than 5,000 labeled instances. In total the dataset has 2,500,000 labeled instances in 328,000 images. In CIFAR-100, there are 100 classes containing 600 images each out of which 500 are training images and the rest for testing, per class. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

Before feeding the data into baseline architectures we performed data augmentation techniques such as image normalization, random crop, MixUp Alpha, color jitter and affine transformation. As ResNet is originally trained on ImageNet dataset whose every image is of size 224x224, therefore, we resized CIFAR images to 224x224 before passing them as input to the ResNet model.

4 Evaluation Metrics

In this project, we will use accuracy to measure the performance of image classifiers and average precision for object detectors. In addition, for comparison of DETR variants, we employ intersection of union which is the fraction of predicted bounding box area over the ground truth bounding box area.

5 Working Baseline

In order to gather information on various architectures, we ran experiments to derive baseline performances. We ran three models on the CIFAR-10 and CIFAR-100 datasets, namely ViT B16-R50, ViT B16 and Resnet101. ViT B16-R50 is a hybrid model comprising pure visual transformer and ResNet50. The Visual Transformer is implemented in the Jax library which harnesses the power of Tensor Processing Units (TPUs). It distributes the traning across 8 TPU cores.

For ViT B16-R50 and ViT B16, we trained the model for 100 epochs, with batch size of 512 and learning rate of 0.03. In case of ResNet101, we used an initial learning rate of 0.01, batch size of 64. In addition, we used a Reduce On Plateau learning rate scheduler with decay factor as 0.1, patience of 3 and a fixed threshold of 0.002.

The validation accuracies for different architectures on CIFAR-10 and CIFAR-100 are depicted in Table 1. From Fig. 1, We observe that the visual transformer has comparable performance with ResNet101 after abour 100 epochs.

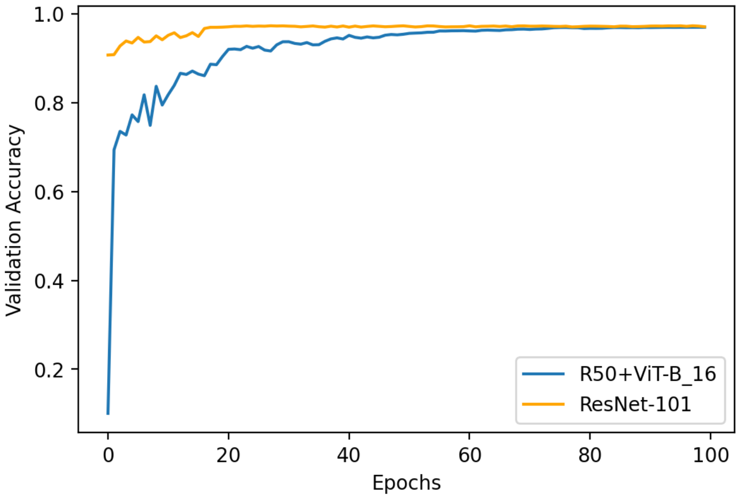

Figure 1: Comparison of CIFAR-10 validation accuracy between hybrid ViT and ResNet101

We performed inference on the COCO dataset using two different DETR backbone architectures, ResNet-50 and ResNet-101. The inference dataset consisted of 2500 images with the metrics as Average Precision and Average Recall that can observed from the Table 2 and Table 3 respectively.

Table 1: Validation Accuracy Percentage

	ViT B16-R50	ViT B16	ResNet101
CIFAR-10	97.337	96.335	97.070
CIFAR-100	73.600	69.973	84.060
Avg Training Hours	6.13	5.36	8.05

Table 2: DETR Performance with ResNet50 Backbone

Backbone	IoU	Area	Avg Precision	Avg Recall
ResNet50	0.5	small	0.205	0.312
	0.5 - 0.9	medium	0.458	0.628
	0.5 - 0.9	large	0.611	0.805
	0.5	all	0.495	0.480

6 Methodology

In this section, we present an overview of the fundamentals of a transformer, the Detection Transformer, the Vision Transformer followed by our modification ideas.

6.1 Transformer

A transformer consists of an encoder and a decoder as shown in Fig.2. The encoder is fed a sequence of representations of text or image $(x_1, x_2, \dots x_n)$ and generates a transformed representation, $z = (z_1, z_2, ...z_n)$. The output of the encoder is fed as input to the decoder which generates an output sequence $(y_1, y_2, ...y_n)$. Within an encoder, there are two functional blocks. The first is a multi-head self-attention block and the second is a fully connected feed forward network. Both the blocks have a residual connection. The decoder has a masked multi-head attention block and a feed forward network. In addition, there is a multi-head attention block whose input is the output of the encoder. Masking is used to prevent the model from peeking at subsequent positions which ensures that the output embeddings at time t are only dependent on known information till time $t - 1$. The encoder and decoder both employ residual connections and normalization layers between functional blocks. The challenge at hand is to convert an image or a set of images into sequences that can be passed as input to the transformer.

A transformer uses positional encodings to gather information on the order of sequences. The relative or absolute position between elements of a sequence must be fed into the network. In case of images, these elements can be individual pixels or subsequences of pixels. The function to generate positional encoding can be learned or explicitly defined for eg. a sinusoid as given in the equation.

$$PE = \sin{(pc^{\frac{2i}{d}})} \tag{1}$$

where p is the position, c is a constant, i is the dimension and d is the dimension of the embeddings.

Table 3: DETR Performance with ResNet101 Backbone

Backbone	IoU	Area	Avg Precision	Avg Recall
ResNet101	0.5	small	0.218	0.337
	0.5 - 0.9	medium	0.479	0.644
	0.5 - 0.9	large	0.618	0.815
	0.5	all	0.512	0.494

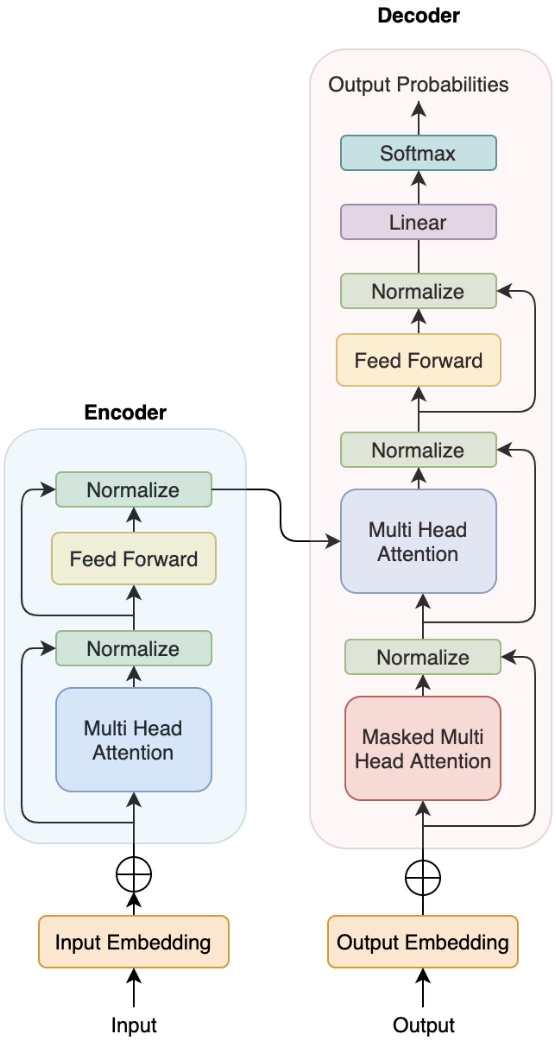

Figure 2: Transformer Architecture

6.2 DETR

The inspiration for our project is derived from the DETR, Detection Transformer. The DETR object detector removes the need for hand-designed feature engineering. It can be viewed as a direct set prediction problem which categorizes the DETR into a fully end-to-end solution. It employs a combination of CNN and transformer. To compute the loss between the bounding boxes of the predicted and ground-truth examples, the authors use a bipartite matching scheme. In case of no match, the prediction is classified as “no object”. This architecture performs at par with highly sophisticated models such as the Faster-RCNN. The nature of bipartite matching loss function allows the opportunity for parallel predictions. Employing the Hungarian loss function for bipartite matching ensures permutation invariance of the predictions.

The DETR cost function and loss function are defined as follows. For every iteration, a fixed value of N objects are detected. The goal of the cost function is to obtain the optimal permutation of predicted objects such as the pairwise matching cost is minimum. Every training example i is defined as $y_i = (c_i, b_i)$ where c is the class and b is a vector that denotes the bounding box. The cost function is given as

$$\hat{\sigma} = \arg\max_{\sigma} \sum_{i}^{N} L_{match}(y_i, \hat{y_i}) \tag{2}$$

where $\hat{\sigma}$ is the optimal permutation, L_{match} is the pairwise matching cost and $\hat{y_i}$ is the prediction. The complete loss function, i.e. the Hungarian loss is defined as

$$L_{hungarian}(y_i, \hat{y_i}) = \sum_{i}^{N} [-\log \hat{p}(c_i) + 1_{c_i \neq \phi} L_{box}(b_i, \hat{b_i})] \tag{3}$$

where $\hat{p}(c_i)$ is the probability of class c_i, L_{box} is the linear combination of $L1$ loss and IoU loss functions.

$$L_{box} = \lambda_{iou} L_{iou}(b, \hat{b}) + \lambda_{L1} ||b_i - \hat{b_i}|| \tag{4}$$

The DETR architecture consists of a CNN backbone which generates feature representations, an encoder, a decoder and a feed forward layer. The input i.e. images have the shape $d \times H \times W$ where d is the dimension, H is the height and W is the weight. The input to the encoder has to be a sequence, therefore, an input image is converted into $d \times HW$ feature map.

For N predictions in an image, N object queries are fed into the encoder and decoder. These object queries are the sum of feature maps and positional encodings. The transformer decoder outputs the class labels and box coordinates that are fed into the feed forward network. As there are N object queries and N outputs from the decoder, therefore, N feed forward networks are used to predict objects which have shared parameters. Each FFN is a multilabel 3-layered multi-layered perceptron with ReLU activation function. This is followed by the bipartite matching loss function which trains the entire DETR. The DETR architecture is shown in Fig. 3.

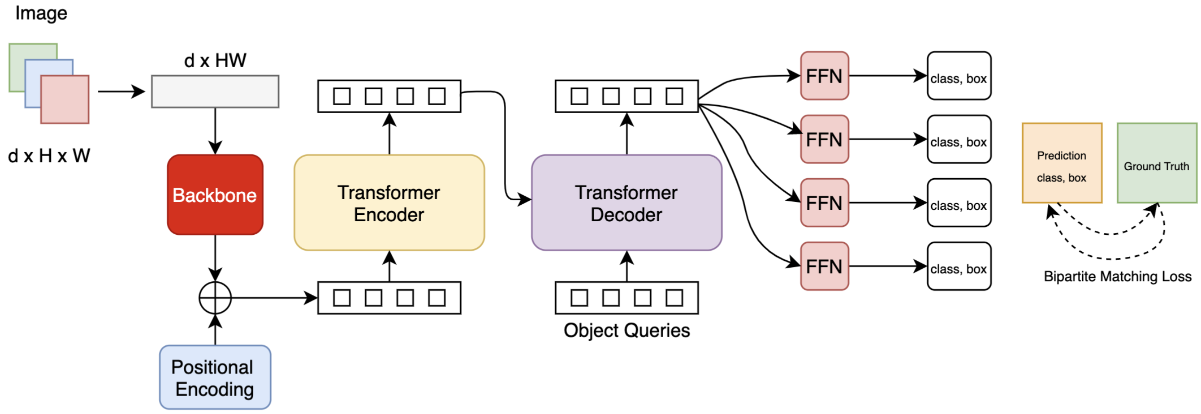

Figure 3: DETR Architecture

6.3 Vision Transformer

The primary objective of transformers here is that they can be applied directly to sequences of image patches and can perform well on classification tasks. With the models and datasets growing, there is still no sign of saturating performance. The Visual transformer is employed to deal with 2D images, where the image $H \times W \times C$ is reshaped into a sequence of flattened 2D patches. These patches are of size $p \times p$ and $N = \frac{HW}{P^2}$ that are the resulting number of patches which will be the sequence length for the transformer. Since the transformer operates on a latent vector size D, the patches are flattened and mapped to D dimensions with a trainable linear projection (learnable weights) which are referred to as the patch embeddings. This is similar to BERT's token where a learnable embedding is prepended to the sequence of embedded patches, whose state at the output of the transformer encoder serves as the image representation y and the position embeddings are added to the patch embeddings to retain positional information.

A more powerful hybrid architecture can be designed by obtaining the raw embeddings from Convolutional Neural Networks wherein the CNN is applied to the entire image and then the feature maps are flattened and projected to the Transformer dimension. In this hybrid structure, the classification input embedding, and the position embeddings are added. For high resolution images, the patch size is kept the same which results in a larger effective sequence length. Although, the transformer can operate on varying sequence lengths, the pretrained position embeddings may not be helpful. Therefore, 2D interpolation is performed of the pre trained position embeddings, according to their location in the original image.

6.4 Detection using Pure Transformers

Exploring the application of transformers in computer vision tasks is still at a fledgling state but is gaining rapid momentum. We have accumulated the baseline architecture performances of ResNet101, hybdrid visual transformer and the DETR. In the next steps, we will modify the backbone of the DETR architecture given in Fig. 3. The backbone of the original DETR utilizes ResNet50 or ResNet101. We hypothesize that replacing the CNN based backbone with a transformer can give competitive results. This modification will lead to a pure End-to-End transformer object detector. This implementation can also prove to be efficient in terms of space complexity, given the significantly less number of training parameters in a transformer compared to a ResNet model.

In addition, we will explore alternatives to the Hungarian loss and bipartite matching in order to improve the time complexity of the divergence calculation between the predicted labels and the ground truth.

Division of Work

Rituraj Singh; experiments with Vision Transformers on CIFAR datasets, literature review, report writing.

Vivek Aswal; experiments with Vision Transformer, ResNet101 backbones, literature review of ViT, RCNN.

Wallace Dalmet; experiments with ResNet50 backbone, bipartite matching and Hungarian loss and literature review.

Tushar Vatsa; experiments with Detection Transformers on COCO dataset with R101 and R50 backbones, literature review on DETR.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

[2] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In European Conference on Computer Vision (pp. 213-229). Springer, Cham

[3] 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.

[4] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv preprint arXiv:2103.14030.

—-

[5] 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.

[6] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[8] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In CVPR, 2018

[9] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In ECCV, 2018

[10] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning RoI transformer for oriented object detection in aerial images. In CVPR, 2019

[11] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In CVPR, 2017.

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In ECCV, 2016.

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

[14] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017

[15] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2325–2333, 2016

[16] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

[17] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. arXiv preprint arXiv:2011.10881, 2020

[18] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. arXiv preprint arXiv:2011.09315, 2020

[19] Gao, P., Zheng, M., Wang, X., Dai, J., Li, H. (2021). Fast convergence of detr with spatially modulated co-attention. arXiv preprint arXiv:2101.07448.

[20] 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, 2020

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve J ́egou. Training data-efficient image transformers distillation through attention. arXiv preprint arXiv:2012.12877, 2020