Place of Attention Matters!

An End-to-End Object Detection with Vision Transformation

Saeed Firouzi
Sharif University of Technology
saeedmr881@gmail.com

June 2023

Abstract

In the object detection task, the purpose is to find the class of object and a bounding box around it. Most works have focused on just finding the class of object without considering bounding box features properly. We present a new method that focuses on relationships between patches of the image as a feature for bounding box detector.
Also, we combine convolutional neural network as a local feature detector and Transformer network as a long-distance feature detector. We were also inspired by the method that has been used in Transformer as a relationship between patches in the image.
Our implementation can perform in real-time and improve the accuracy of previous works.
The training code and pre-training model are available at
https://github.com/saeed5959/object_detection_transformer
to help open source community.

1 Introduction

Most works has shown that convolutional networks are basically considered as local feature detectors. It is quite easy to see if we are familiar with CNN architecture. Using a filter with small dimensions like : 3*3 , 5*5 , … , 9*9 can just get a local feature around a pixel value. The strength of using CNN is that we produce a high number of these filters to capture features with different styles. Some attempts like max-pooling, deformable convolution, dilated casual convolution have been applied to increase receptive fields of a pixel, to look further and to see more around it. But still they have not overcome this issue.

On the other hand, after introducing Transformer, and specifically vision transformer, many works have been tried to apply transformer to image input to achieve a long-term dependency between pixels or patches of image. The main idea behind the transformer is to get a weighted sum for a vector by point-wise multiplying of vectors as a similarity term.
That means by giving a vector to a transformer, we will get a vector with the same dimension as the input but some values related to a feature have been increased or decreased based on whether that feature exists in other vectors or not.

We were inspired by this point from transform, to use it for bounding box feature to be our major contribution to this work. We call it place of attention matrix (POA matrix) to be a feature guide for bounding box detection.

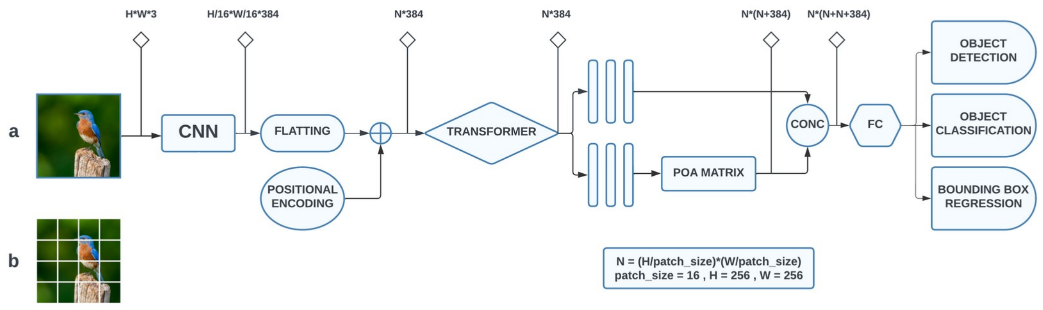

Figure 1. **An illustration of model architecture (POA-ViT).** In (a) we pass the image to the CNN network. After patching and flatting, we give it to the transformer block. Then concatenate it with POA matrix and using FC layer, we can get the existence of object and class and bounding box for any patch. In (b) we show an image that has been divided to 4*4 patch.

Mostly there are two kinds of network: two-stage and one-stage networks. In two-stage networks, they mostly use a region-proposal-network(RPN) for suggesting some box for objects. These boxes are predefined and have different ratios and aspects. Ratios are from 20,50,100,200 pixels in size and aspects are from 1:1, 1;2, 2;1, … size. In most cases, the number of these boxes becomes more than 2000 boxes. These proposal boxes have 2 drawbacks; first, they need a lot of computation, which makes it impossible for real-time tasks, and second, some objects can not be fit in these predefined boxes.

In a one-stage network, the main idea is to predict the bounding box coordinate directly. By dividing the input image into patches and predicting a bounding box for these patches, it is possible to make the network work in real-time tasks.

In this method, we will use a one-stage network, and divide the input image into patches and pass patches to the network that consisted of CNN and Transformer to get a feature vector for every patch. Then we will calculate POA matrix by multiplying a feature vector of a patch on other vectors to get a similarity term as a factor of how much this patch is similar to another patch. And then concatenate this POA matrix with feature vector and then apply a linear layer to get a score for objectness and 90 scores for 90 class and 4 values: x of center, y of center, width and length as bounding box parameters.

Our main contributions is :

- Defining POA matrix to help object classification and bounding box regression
- Applying a new strategy for training by using segmentation dataset to avoid from misinformation

2 Model Architecture

2.1 CNN Network

Instead of giving raw pixel value to network and just applying FC layer [1] that can’t get local features properly, we use a convolutional network by inspiring from ResNet [2] architecture. The main idea in ResNet is using skip-connection to pass data directly into deeper layers. Also, we use max pooling in our block as a dimension reduction to decrease the resolution of input 16x smaller.

As a regularization factor, we use both batch normalization and layer normalization due to a different variety of input images.

Also in the last block we use 9*9 kernel filter to increase receptive fileds.

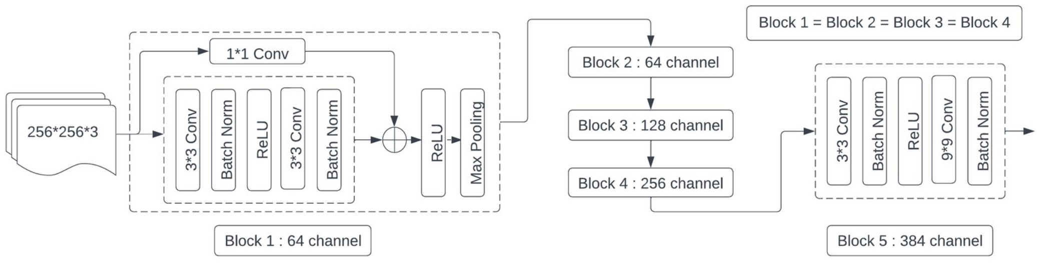

Figure 2. **The CNN network.** Last block uses 9*9 filter to increases receptive filed.

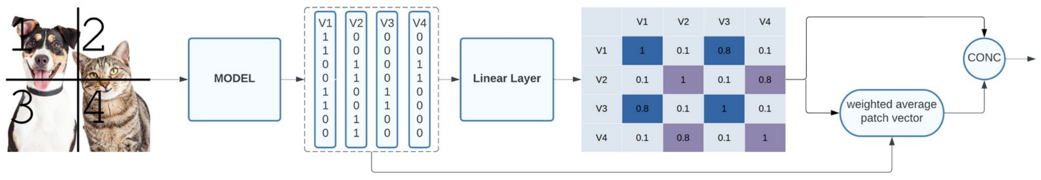

Figure 3. **POA matrix.** For an image with 4 patches, for every patch we have a feature vector; then the task of POA matrix would be to calculate similarity between patches by point-wise multiplying of feature vectors. This helps the model to know which patches belong to an object, then the model can predict the bounding box of object properly. In this case, patch 1,3 has similar features, then they belong to an object and also the 2,4 patches. Also by averaging all patch vector with weights of POA matrix, we can have a average vector for an object. This helps the model to classify every patch properly.

2.2 Patching and Flatting

After applying CNN network in input image with size H*W with 3 channels as RGB values, we will get data with size H/16*W/16 with 384 channels as a feature values.
Now, inspired by YOLO architecture [3], we divide this into the same size patches and then flat them in raster order to get a feature vector for every patch.

3.3 Transformer

Now we have N patches with size 384. In Transformer we add positional encoding to feature vectors to give spatial information to the model for bounding box prediction. This positional encoding is an embedding layer, giving input of (0, number of patches) in a raster order to be learnable. If we consider a feature vector as a container of features of different objects, then Transformer’s task is to increase values of some objects in this feature vector with respect to other patch feature vector.
We use multi-head attention with a sequence of linear layers [4], and giving feature vector to query, key, value as inputs of Transformer block.

3.4 Place of Attention Matrix (POA Matrix)

By dividing the image into small patches, we can capture small objects. But for large objects, we should make a relationship between patches. By point-wise multiplying every patch vector with other patch vectors, we can get a similarity and positional relationship between them. Then we have a matrix that we call it the place of attention matrix (POA matrix). For every patch vector, we concatenate its corresponding row of POA matrix and also we can get a weighted average of all vectors as a average object vector. First by concatenating corresponding row of POA matrix it can help to improve bounding box and second by concatenating weighted average of vectors it can help for classifying object correctly. By concatenating these three vectors and giving it to linear layer it can detect objectness and classify the patch and get the bounding box coordinate if it does not belong to the background class. This method is inspired by attention in transformer[4].

4 TRAINING

4.1 Training Strategy

Despite YOLO [3], which makes the center patch responsible for classifying the object and bounding box coordinate, and despite Faster-RCNN [5], which makes all pixels in the box

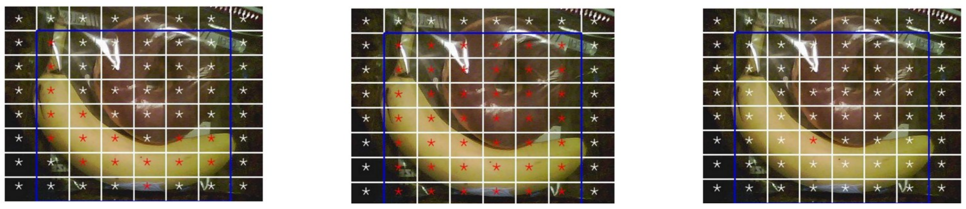

Figure 4. **Illustration of different training methods**. in (a) we just calculate the loss for red-star patches that contain objects. In (b) it uses all pixels of bounding box. As you can see, some pixels and patches don't contain the object and are just misinformation. In (c) it just uses one center patch and ignore some other patches that contain objects.

responsible for object detection, we use segmentation data beside the object box, and instead of making all patches inside of box responsible for object detection, just use patches that have intersection with segmentation data.
This approach helps to ignore misinformation that may come from patches inside the box that does not contain the object.

4.2 Augmentation

Instead of pre-training the model on the ImageNet dataset, we use the augmentation method to increase the number of images for better accuracy.
For augmentation, we use shift, scale, rotation and also the Cutout method[6] to force model to learn different features of an object.

4.3 Loss

Model outputs (1+C+4) values; 1 for objectness and C for number of classes and 4 for bounding box values. For objectness we use binary classification and for classification we use crossentropy loss and for bounding box we use L1 loss.
We consider Bounding box values as a relative parameters to the whole image, from (0 to 1).
Also we consider a POA loss for POA matrix. This consists of 1 for patches of same object and 0 for non related patches. The model use teacher forcing method for a limited number of epoch to use ground-truth POA matrix for training.

5 EXPERIMENT

We train the model on COCO dataset [7] that includes 118,000 images with 90 categories.
Every image contains a bounding box for objects and a segmentation image.
We evaluate our model based on two factors; accuracy and inference time.
We use a batch size 32 and train on V-100 gpu.

	MAP(0.5)	Recall	Inference time (fps)
POA-ViT	52%	30%	5

6 Conclusion

we introduced POA-ViT model as a unified and one-stage model that can inference in real-time speed and has a acceptable accuracy. Our main contribution is to introduce POA matrix and a new strategy for training by using segmentation dataset to avoid form misinformation.
Our model is fast and can detect objects in different shape and size accurately.

7 further research

The main focus for further research will be to use a pyramid method to capture data in different stages and size.

References

[1] 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.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 10 Dec 2015

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, 9 May 2016

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention Is All You Need, 6 Dec 2017

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 6 Jan 2016

[6] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with CutOut. arXiv preprint arXiv:1708.04552, 2017

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), pages 740–755, 2014