Place of Attention Matters!

An End-to-End Object Detection with Vision Transformation

Saeed Firouzi
Sharif University of Technology
saeedmr881@gmail.com

Abstract

In the object detection task, the purpose is to find the class of object and a bounding box around it. Most of works has focused on just finding the class of object without considering bounding box features properly. We present a new method that focus on relationship between patches of 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 also inspire the method that has been used in Transformer as a relationship between patches in image.
Our implementation can perform in real-time and improve the accuracy of previous works. Training code and pre-training model are available at
https://github.com/saeed5959/object_detection_transformer
to help open source community.

1 Introduction

Most of works has shown that convolutional network are basically consider as a local feature detector. It is quite easy to see if we be familiar with CNN architecture. Using a filter with small dimension 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 high number of these filters to capture features with different styles. some attempts like max-pooling, deformable convolution, dilated casual convolution has 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.

In the other hand, after introducing Transformer, and specifically vision transformer, many works has been tried to apply transformer to image input to get 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 same dimension as input but some values related to a feature has been increased or decrease based on that features is also exists in other vector or not.

We inspired 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.

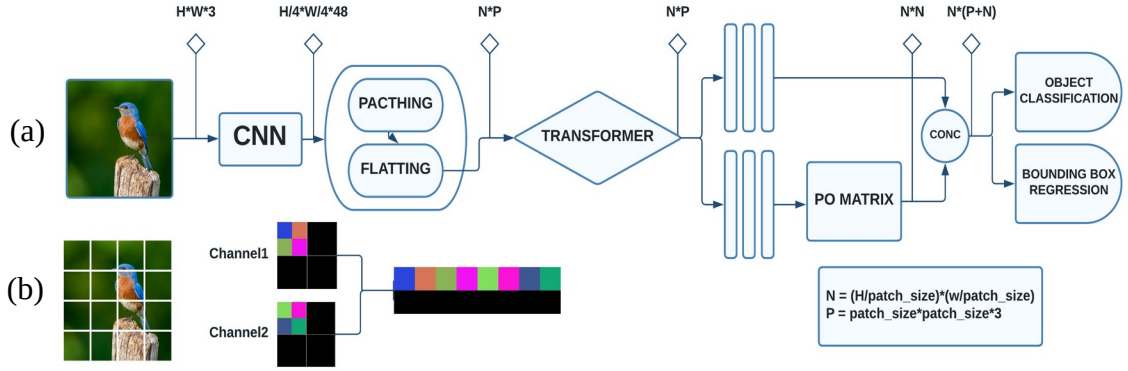


Figure 1. **An illustration of model architecture (PoA-ViT).** In (a) we pass image to cnn network. After patching and flatting we give it to transformer block. Then concatenate it with POA matrix and using 2 linear layer we can get class and bounding box of object. In (b) we show how we patch and flat output of cnn network to make it one dimension vector for transformer.

Mostly there are two kind of network: two-stage and one-stage networks. In two-stage networks mostly they use a region-proposal-network(RPN) for suggesting some box for objects. These box are predefined and have different ratio and aspect. Ratio are from 20,50,100,200 pixels size and aspects are from 1:1, 1;2, 2;1, … size. In most cases the numbers of these boxes become more than 2000 boxes. This proposal boxes has 2 drawbacks; first it needs a lot computation that makes it impossible for real-time tasks and second, some object can not be fit in these predefined box.

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

In this method, we will use one-stage network, and divide the input image to patches and pass patches to the network that are 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 patch at other vectors to get a similarity term as a factor that 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 class and another linear layer that outputs 4 values: x of center, y of center, width and length as bounding box parameters.

2 Model Architecture

2.1 CNN Network

Instead of giving raw pixel value to network and just applying FC layer [Detr] that can't get local feature properly, we use a convolutional network by inspiring from ResNet [] architecture. The main idea in ResNet is using skip-connection to pass data directly in deeper layers. Also we just use 2 max pooling in our block, to keep spatial features for model to detect location of features in image. because it can decrease resolution of input 4x smaller.

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

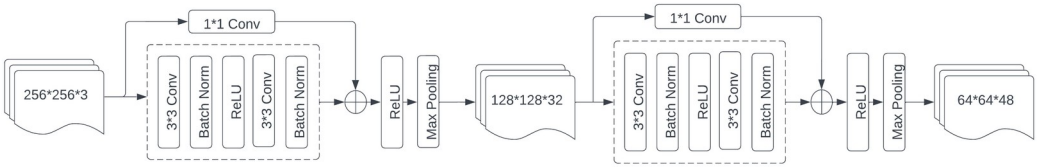


Figure 2. **The CNN network**

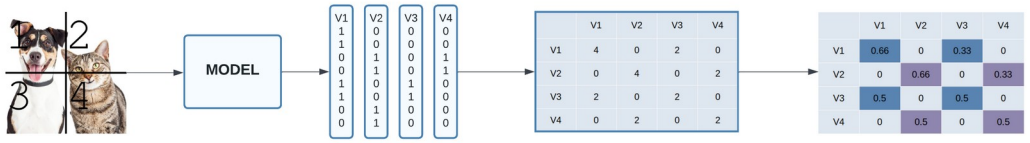


Figure 3. **POA matrix.** If we have a image with 4 patch and for every patch we have a feature vector, then task of POA matrix would be to find similar patch by point-wise multiplying of feature vectors to get a similarity factor. This helps the model to know witch patches are belong to an object then the model can predict bounding box of object properly. In this case patch 1,3 has a similar features then they are belong to an object and also for 2,4 patches.

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 a data with size H/4*W/4 with 48 channels as a features values.
Now with inspiring from YOLO architecture [], we divide this to same size patches and then flat it in raster order alongside of it's channels to get a feature vector for every patch. See Figure 1.b.

3.3 Transformer

Now we have N patches with size P. In Transformer we add positional encoding to feature vector to give spatial information to the model for bounding box prediction. This positional encoding is a embedding layer with giving input of (0, number of patches) in a raster order to be learnable. If we consider feature vector as a container of features of different objects, then Transformer task is to increase values of some objects in this feature vector with respect to other patch feature vector, that there is a object in that patch or not.
We use multi-head attention with sequence of linear layer [], and giving feature vector to query, key, value as inputs of Transformer block.

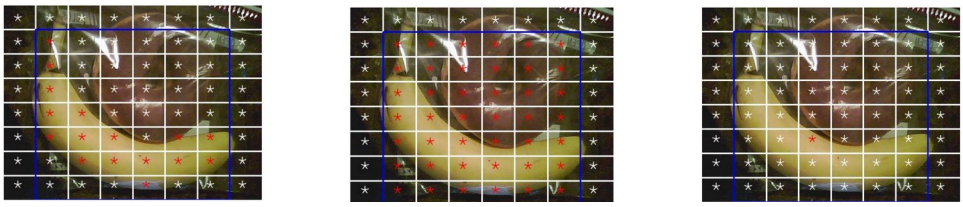
3.4 Place of Attention Matrix (POA Matrix)

By dividing image to small patches, we can capture small objects. But for large objects we should make a relationship between patches. By point-wise multiplying every patch with other patches we can get a similarity and positional relationship between them. Then we have a matrix that we call it place of attention matrix (POA matrix). For every patch vector, we concatenate its corresponding row of POA matrix and by giving it to 2 linear layer, we can classify the patch and get bounding box coordinate if it is not belong to background class. This method is inspired by attention in transformer[].

4 TRAINING

4.1 Training Strategy

Despite of YOLO [], that makes the center patch responsible for classifying the object and bounding box coordinate, and despite of Faster-RCNN [], that makes all pixels in box responsible for object detection, we use segmentation data beside of object box, to instead



a) Our strategy b) Faster-RCNN c) YOLO

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

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

4.2 Augmentation

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

5 EXPERIMENT

We train the model on COCO dataset [] that contains 118,000 images with 90 categories. Every image contains a bounding box for objects and a segmentation image.
We evaluate our model based on 2 factor; accuracy and inference time. Then we will compare our model with Faster-RCNN and YOLO-V5 and Detr.