Single Stage Object Detectors*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Jyothi Vishnu Vardhan kolla
Khourgy college of Computer Science
Northeastern University
Boston, USA
kolla.j@northeastern.edu

2nd Karan Shah
Khoury college of Computer Science
Northeastern University
Boston, USA
shah.karan3@northeastern.edu

***Abstract*—In the rapidly advancing field of computer vision, object detection stands out as a critical component with significant real-time applications. This project aims to underscore the pivotal role of object detection in contemporary technological contexts, particularly highlighting its utility in real-time scenarios. Central to our exploration are two cutting-edge methodologies: You Only Look Once (YOLO) and DEtection Transformers (DETR). These methods are renowned for their simplicity, elegance, and efficiency, marking a paradigm shift in how object detection is approached and implemented. YOLO, with its unique single neural network architecture, offers a swift and effective means of detecting objects in real time, making it highly suitable for applications requiring immediate response, such as autonomous driving and surveillance systems. On the other hand, DETR, leveraging the power of transformers, introduces a novel approach by eliminating the need for many hand-designed components, streamlining the process of object detection. This project not only delves into the technical intricacies of these methods but also empirically demonstrates their effectiveness through a series of experiments and applications. The results obtained underscore the profound impact and efficiency of YOLO and DETR in real-time object detection, paving the way for more advanced and intuitive computer vision solutions in the future.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Object detection, a fundamental task in the field of computer vision, involves not only recognizing a variety of objects within an image but also determining their location and size through bounding boxes. This differs significantly from object classification, which is solely concerned with identifying the presence of an object in an image. While classification answers the 'what' in an image, detection addresses both 'what' and 'where', making it a more complex and nuanced task.

The evolution of object detection methodologies illustrates a trajectory of growing sophistication. Initially conceptualized as simple localization, early techniques involved augmenting linear layers to classification models to predict bounding boxes. This approach, though rudimentary, laid the foundation for more advanced developments. Subsequently, window-based methods emerged, enhancing the capability to manage multiple objects within a single frame. These methods scanned images with various window sizes to identify and locate objects, albeit with considerable computational demands.

The field then witnessed a significant leap with the introduction of methods like R-CNN (Region-based Convolutional Neural Networks) and its more efficient successor, Fast R-CNN. These methods innovatively combined region proposals with deep learning, drastically improving accuracy but at the cost of speed and computational resources. The quest for efficiency and precision continued, leading to the advent of more advanced techniques.

The emergence of single-shot methods like YOLO (You Only Look Once) and DETR (DEtection TRansformer) marked a paradigm shift in object detection. YOLO streamlines the process by using a single neural network to simultaneously predict bounding boxes and class probabilities, dramatically increasing speed and suitability for real-time applications. DETR, on the other hand, employs the transformer architecture to treat object detection as a direct set prediction problem, eliminating many hand-designed components of previous systems. This not only simplifies the detection pipeline but also enhances accuracy, particularly in complex scenarios with densely populated objects.

These cutting-edge methodologies, epitomized by YOLO and DETR, represent not just incremental improvements but significant strides in object detection. They underscore a move towards more intuitive, efficient, and accurate systems, paving the way for groundbreaking real-time applications and future research in computer vision.

II. Related Work

In the realm of object detection, classical methodologies traditionally followed a multi-step approach. Initially, images were subjected to thresholding to create binary representations, which were then refined through morphological filtering, involving 4-connect erosion and 8-connect dilation operations. Subsequently, these processed images were segmented into distinct regions using techniques like the two-pass connected components algorithm. Each of these regions was analyzed to extract valuable features, such as moments, height, and width,

Identify applicable funding agency here. If none, delete this.

in addition to generating bounding boxes. The classification of these regions was determined by comparing their features to those from a training dataset, employing various distance metrics.

As the field evolved, a shift towards convolutional neural networks (CNNs) took place, allowing for more efficient and accurate object detection. Initially, CNNs like AlexNet and VGG were employed for holistic object recognition and classification, benefiting from fixed input and output sizes. However, the task of object detection, which involves identifying multiple objects within an image, each with its bounding box and class label, necessitated the development of specialized architectures.

This led to the emergence of two distinct families of deep networks. The first family comprises Region-based CNNs (RCNNs), known as two-stage networks. In the initial stage, these networks generate region proposals for the input image, and in the subsequent stage, these regions undergo further processing through a CNN to obtain both object classifications and bounding box coordinates. RCNNs typically employ classical computer vision components, such as selective search, to compute region proposals. It's important to note that RCNNs are not end-to-end solutions, unlike Faster RCNN.

In contrast, the You Only Look Once (YOLO) approach takes a novel one-stage approach. YOLO divides an input image into an S x S grid and processes the entire image through a single CNN network. This single-pass operation yields predictions for bounding boxes and class labels, making YOLO a highly efficient and real-time object detection method.

In recent years, the computer vision community has witnessed a groundbreaking shift in object detection methodologies with the introduction of the DEtection Transfomer (DETR). DETR represents a departure from the traditional two-stage and one-stage approaches, offering a novel perspective by leveraging the power of Transformers, originally developed for natural language processing tasks.

These advancements represent significant strides in the field of computer vision, offering a range of approaches that have reshaped the landscape of object detection, from traditional methods to the state-of-the-art deep learning techniques like Faster RCNN and YOLO.

III. Methodology

A. YOLO

In our object detection approach, we try to reproduce YOLOv1 (You Only Look Once version 1) from scratch. Specifically, following the paper, our YOLOv1-based architecture divides the input image into an S x S grid, which is then processed through a convolutional neural network (CNN) backbone. For each grid cell, our model predicts B bounding boxes' coordinates (x, y, w, h), confidence scores for these bounding boxes, and class probabilities for C different object classes. These predictions are structured as an S x S x (B x 5 + C) tensor. The bounding box coordinates (x, y) are defined relative to the grid cell, while (w, h) are relative to the entire image. Confidence scores represent both the model's confidence that the box contains an object and the accuracy of the box prediction. Formally, confidence is computed as the product of Pr(object) and the Intersection over Union (IOU) between the predicted and ground truth boxes. To adhere to the YOLOv1 paper's specifications, we set S=7, B=2, and C=20, aligning with the 20 classes present in the PASCAL VOC dataset. Consequently, each image's predictions are encoded as a 7 x 7 x 30 tensor. Our CNN backbone consists of 24 convolutional layers, followed by 2 fully connected layers following the architecture in the paper shown below. All layers, except the final one, employ the leaky Rectified Linear Unit (ReLU) as the activation function.

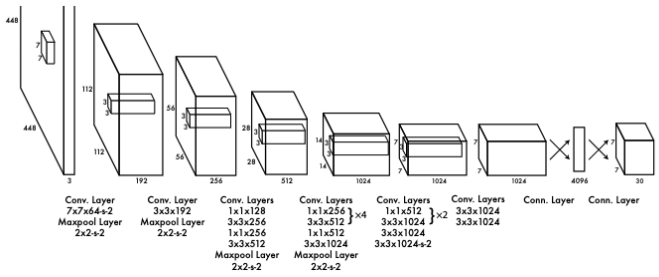

Fig. 1. The Architecture of Yolo

In the realm of object detection, the formulation of the loss function is pivotal for the efficacy of the model. The loss function in our project comprises five distinct components, each tailored to address specific aspects of the detection task. First, the bounding box coordinates, represented by (x, y), are parametrized as offsets from a specific grid cell location, ensuring values range from 0 to 1. The Sum of Squared Errors (SSE) is employed here, but only when an object is actually present within the grid cell.

The second component deals with the bounding box's width and height (w, h). To maintain consistency, we normalize these dimensions by the image's overall size, constraining them to fall between 0 and 1. Here too, SSE is used, but uniquely, we compute it using the square root of the bounding box's width and height. This approach effectively addresses the disproportionate impact of small deviations in larger boxes.

The third and fourth components focus on confidence scores based on Intersection Over Union (IOU). A common challenge in object detection is the imbalance in confidence scores, especially in grid cells devoid of objects. To counter this, the third term reduces the loss contribution from

confidence predictions for boxes without objects by setting lambda noobj to 0.5. The fourth term complements this by further penalizing confidence predictions for empty cells, thus enhancing the model's stability in distinguishing between cells with and without objects.

Lastly, the class probabilities aspect of the loss function is calculated using SSE, but only in instances where an object is present. To underscore the importance of accurate object localization, we apply lambda coord=5, significantly amplifying the loss from bounding box coordinate predictions. This multifaceted approach to the loss function ensures a balanced and effective model, adept at both identifying and accurately localizing objects within an image.

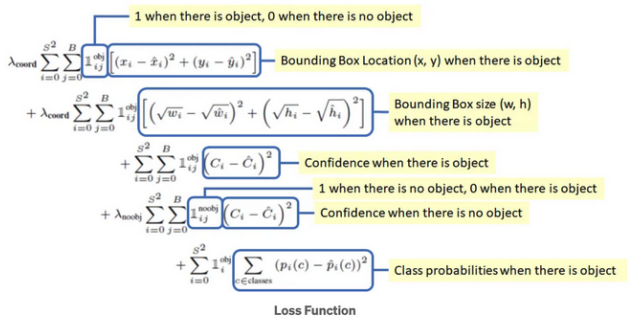

Fig. 2. Loss function of Yolo

In our implementation, we were unable to pretrain for several weeks on ImageNet. In addition, our dataset comprises of the training and validation sets from PASCAL VOC 2007 and 2012 (16000 images). After combining the dataset, we split the data with approximately 13000 images for training and 3000 for validation. Additionally, we employ non-maximum suppression on the predicted bounding boxes for each image to refine the final object detections similar to the paper. We also left code to use ResNet as the backbone architecture; however, we were unable to train this portion due to time constraints.

B. DETR

The Detection Transformer (DETR) represents a paradigm shift in object detection by treating the task as a direct set prediction problem, effectively eliminating the need for many hand-designed components such as anchor generation and non-maximum suppression (NMS) that are common in conventional object detection frameworks.

DETR employs a three-part architecture. The first part is a convolutional neural network (CNN) backbone that processes the input image to generate a lower-resolution activation map. This backbone, which typically uses a ResNet architecture, reduces spatial resolution to create a compact feature representation. The second part is a transformer encoder-decoder, borrowed from the NLP domain. The encoder takes the feature map, flattens it, and then, through a series of self-attention and feed-forward layers, produces an encoded feature representation. Each layer of the encoder and decoder uses multi-head self-attention and includes positional encodings to maintain spatial information. The decoder transforms object queries learned embeddings that represent a fixed-size set of N potential objects into output embeddings. Unlike traditional autoregressive models, DETR decodes all objects in parallel, leveraging global context to enhance prediction accuracy. The third and final part is a feed-forward network (FFN) that outputs the predictions for bounding box coordinates and class labels for each of the N object queries.

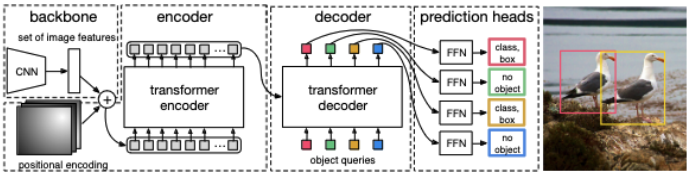

Fig. 3. The Architecture of DETR

The direct set prediction approach in DETR streamlines the detection pipeline. Instead of predicting object locations and classes relative to predefined anchors or regions, DETR outputs a fixed-size set of predictions in parallel, considering the global image context. This method requires the model to learn a set of object queries that correspond to potential objects in the image. The transformer's attention mechanism enables the model to differentiate between these queries without prior knowledge about object size, aspect ratio, or number.

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right),$$

Fig. 4. equation-1

The training process employs a bipartite matching loss (equation-1) to score the predicted set against the ground truth. This loss function ensures a one-to-one matching between predicted and actual objects, addressing permutation invariance inherent in set prediction tasks. To find the optimal matching, the Hungarian algorithm (equation-2) is used, minimizing a cost function that combines class prediction accuracy and bounding box similarity. Once the optimal matching is established, the Hungarian loss is computed as a sum of the negative log-likelihood for class prediction and a box loss for each matched pair. The box loss (equation-3) is

a combination of the L1 loss and the generalized IoU loss, which is scale-invariant. These losses are normalized by the number of objects to address class imbalance, with a higher penalty applied to incorrect predictions for actual objects compared to empty slots, represented by a special 'no object' class in the set of predictions.

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

Fig. 5. equation-2

DETR's innovative architecture and its direct set prediction approach, combined with the defined loss functions, enable end-to-end object detection that is both efficient and highly effective, showing promising results on standard benchmarks like the COCO dataset.

RESULTS

In our implementation of YOLOv1, we conducted training for approximately 25 epochs, adhering to the general procedure outlined in the original paper. The training process involved optimizing the network's parameters to learn object detection tasks. The progress of the training is illustrated in the figure below, which presents Mean Average Precision (MAP) metrics for both the training and validation datasets.

After 25 epochs, the training MAP reached an approximate value of 0.67, showcasing the model's capability to fit the training data. However, a noteworthy observation is that the validation MAP remained significantly lower, at around 0.1. This discrepancy between the training and validation MAP suggests a clear case of overfitting, where the model's performance on the training data surpassed its generalization to unseen validation data.

It is essential to highlight that in the original YOLOv1 paper, the authors conducted extensive training, involving approximately 135 epochs on their dataset, in addition to pre-training on the ImageNet dataset for an entire week. Unfortunately, due to constraints in computational resources and time limitations, our training was limited to a shorter duration of 7-8 hours on a V100 GPU.

The substantial performance gap between the training and validation MAP indicates that further training and regularization techniques are necessary to improve model generalization. Future work could involve exploring techniques such as data augmentation, dropout, or early stopping to mitigate overfitting and enhance the model's ability to perform well on unseen data.

In summary, our experiments reveal that, while YOLOv1 demonstrates promising results on the training data, additional training epochs and regularization strategies are required to bridge the performance gap between the training and validation datasets and enhance the model's ability to generalize effectively.

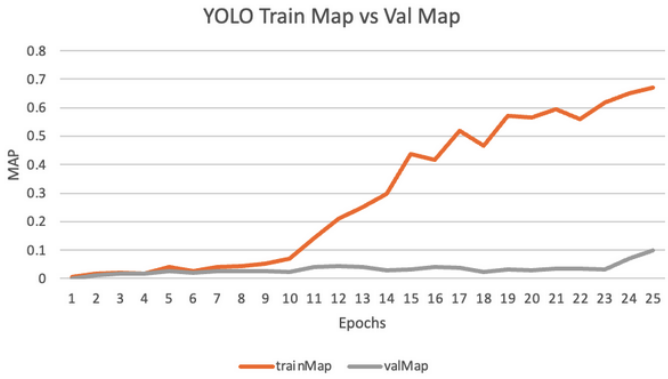

Fig. 6. train map vs val map

In our implementation of the Detection Transformer (DETR) model using PyTorch, we adhered closely to the methodology outlined in the original paper. Due to computational limitations, we opted to initialize the model's weights with pre-trained parameters made available by Facebook. This approach allowed us to bypass the computationally intensive training phase, leveraging the robustness of weights already optimized on a substantial dataset. Subsequently, we deployed the model on our test dataset, where we achieved an MAP of 0.42 on COCO 2017 val set.

Some of the detections given by DETR on our test dataset:

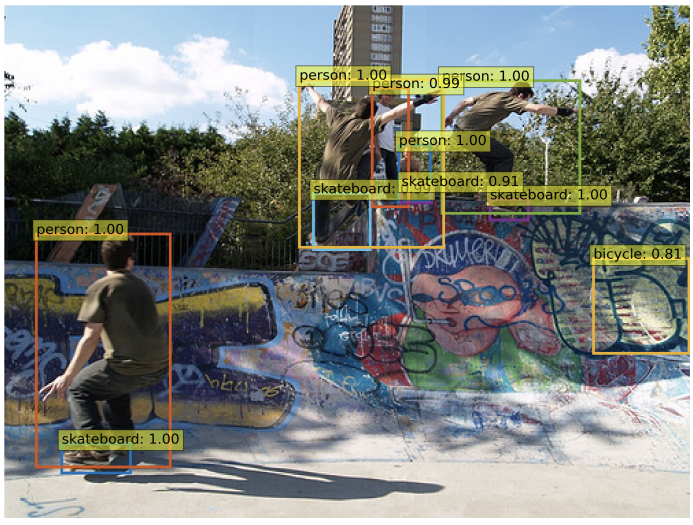

Fig. 7. figure-1

DISCUSSIONS AND SUMMARY

YOLO remains widely used today and has influenced subsequent object detection architectures. Its real-time

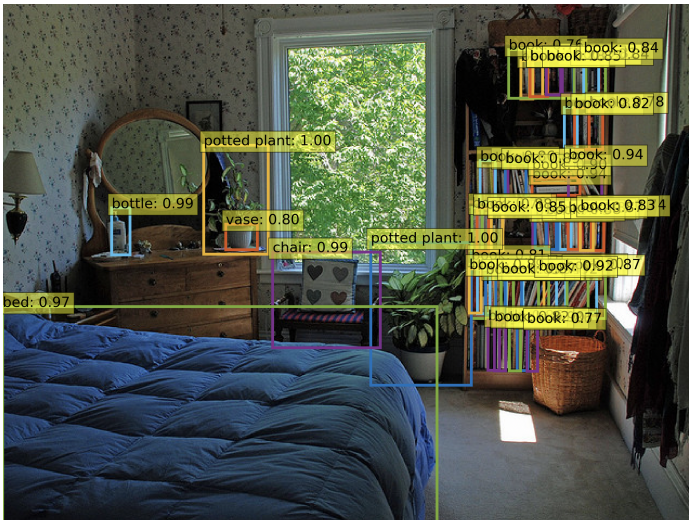

Fig. 8. figure-2

Fig. 9. figure-3

capabilities are essential in applications like surveillance cameras, object tracking, and robotics. Researchers and developers continue to build upon and optimize the YOLO framework with current SOTA being YOLOv8. However, from our experimentation, YOLO still took a considerable amount of time to train but that can also be attested to the fact that we worked with such a large dataset of images. On the other hand, one of the biggest drawbacks of YOLO is that it can only classify S^2 objects and in this case since S=7, it can only classify 49 objects within an image.

DETR has ushered in a new era of object detection by showcasing the effectiveness of Transformers in computer vision tasks. While it may not replace YOLO entirely, DETR's impact is evident in research and industry, pushing the boundaries of what's achievable in object detection.

Fig. 10. figure-4

REFERENCES

[1] Alex Krizhevsky, "One weird trick for parallelizing convolutional neural networks", 1404.5997, 2014.

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

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)", arxiv 2013.

[4] Ross Girshick, "Fast R-CNN", arxiv 2015.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", arxiv 2017.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, "AN IMAGE IS WORTH 16X16 WORDS:TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE", ICLR 2021.

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", arxiv 2016.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-End Object Detection with Transformers", arxiv 2020.