Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

Курсова робота
За спеціальністю “Прикладна Математика”

Object feature extraction for YOLO detectors

Керівник курсової роботи
Ст. Викл. Швай Н.О.
_____ (підпис)
“____” _____ 2023 р.
Виконав:
студент МП-1
факультету інформатики
Абашкін О.В

Table of contents

Table of contents

1. Introduction ……2
2. Theoretical part ……..2
   2.1 YOLOv1……2
   2.2 YOLOv2……4
   2.3 YOLOv3……6
   2.4 YOLOv4……7
3. Experiments ……10
4. Conclusion ……11

1 Introduction

Object detection task is one of the main tasks in the Computer Vision domain, and the ability to make high-quality predictions with high inference speed was not easy to reach, due to approaches that were popular at that time. But the invention of the YOLO algorithm and one-stage approach allowed us to complete this challenge.

2 Theoretical part

2.1 YOLOv1

In this part will be described the key points and the main idea of the paper: You Only Look Once: Unified, Real-Time Object Detection by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi.

The main goal of the research: To create an architecture that can surpass in quality and speed the solutions of that time such as the deformable part models (DPM) that were using the sliding window approach where the classifier is used for each evenly spaced location, and a the R-CNN that were using a network for generation potential bounding boxes and as a second stage applies a classifier on this regions.

The main idea of the YOLO architecture.

The main idea of the YOLO architecture was to get rid of the two-stage processing and create a single neural network that can cover the feature retrieval and bounding box coordinates prediction part.
Main key points:

- Proposed system was dividing the input image in the S x S grid. When the center of the object falls into the cell, this cell will be responsible for detecting that object.

- Each grid cell predicts *B* bounding boxes and confidence scores for these boxes. Confidence is computed by this formula: $Pr(Object) * IOU_{pred}^{gr\ truth}$. If the cell is empty confidence score should equal zero.

- Grid cell predicts *C* conditional class probabilities $Pr(Class_i|Object)$. This score encodes the probability of presence object of the i-th class in the cell.

- The bounding box includes five predictions: $(x, y, w, h, confidence)$. (x, y) coordinates denote the center of the bounding box relative to the bounds of the grid cell.

Network architecture.

The architecture of the YOLO model was inspired by the GoogLeNet by Google that was used for image classification, but instead of using the inception modules authors used the 1x1 size reduction layers followed by the 3x3 convolutional layer.

The main part of the model that is responsible for feature extraction consists of 24 convolutional layers (in the base version) followed by the 2 fully connected layers. The fast YOLO model has 9 convolutional layers with a smaller amount of filters.

For the training was used the leaky-relu function was:

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

Figure 1: Formula of the Leaky ReLu activation function.

And loss function -

Regression loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Classification loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

Figure 2: Formula of the YOLO loss function.

Improvements in further versions.

2.2 YOLOv2 or YOLO9000.

Original YOLO has suffered from a couple of imperfections, such as a significant amont of localization errors and relatively lower recall than region proposal-based models such as Fast R-CNN. YOLOv2 was created to solve these issues, to achieve this goal authors have added new technologies and have changed the existing approaches.

- **Batch Normalization.**

  By adding the Batch Normalization on all convolutional layers in the model authors achieved a 2% percent higher mAP, and the ability to simplify the network by removing the dropout and using Batch Normalization as the one regularization technique. Also, the usage of Batch Normalization leads to significant improvements in convergence.

- **High-Resolution Classifier.**

  The authors doubled up the original input size of the image. And in

YOLOv2 it reached the size of 448x448. This change gave a 4% growth of the mAP.

- **Convolutional with Anchor Boxes.**

  The original YOLO was predicting the bounding box coordinates by using the fully connected layers after a set of the convolutional layers, on the other hand, Region-Proposal Networks predict the offsets and confidence of the anchor boxes, which is much simpler for the network to learn. Thus, the authors decided to reuse this experience, they removed the fully connected layer from the architecture and used the anchor boxes. In addition, one global pooling layer was removed to make the resolution of the output feature map higher. In the second step, the input size was shrunk from 448x448 to 416x416 to get an odd number of cells in the future map, it is caused by the fact that bug object tends to occupy the center of the picture, then it is better to have the one central cell instead of four.

- **Dimension Box Cluster.**

  Anchor boxes have hyperparameter - box dimensions. The network can learn an appropriate way how to define box dimensions but with better priors for the network to start with, the network will learn much easier how to predict good detections. Thus, the authors decided to apply a k-means clustering algorithm on bounding boxes from a training set, to automatically define good priors.

  The function that was used as distance function:

  $d(box,\ centroid)\ =\ 1\ -\ IOU(box,\ centroid)$.

- **Direct location prediction.**

  The second drawback of the anchor boxes is that it leads to model instability, especially in early iterations. Most of the instability was caused by the prediction of (x, y) the location of the box. It is caused by the imperfection of the formula for the prediction, the formula is not

constrained so any anchor box can end up at any point of the image. Thus, the authors decided to predict coordinates relatively to the grid cell. This change with the usage of dimension clusters gave 5% of improvement.

- **Fine-Grained Features.**

  The YOLO algorithm uses a relatively small feature map for using predictions, to be exact, with a size of 13x13, while it is sufficient for detecting large objects it usually is not enough for small objects.
  To solve this issue, the authors decided to add the passthrough from the earlier convolutional layer with a size of 26x26. The pass-through concatenates the high-resolution and low-resolution features by stacking adjacent features into different channels, similar to the identity mapping in the ResNet.

2.3 YOLOv3

During this iteration general approach has not experienced significant changes, except for the new feature extractor, the authors used a new network - Darknet53.

Structure of feature extractor:

Darknet53 it is an improvement of the previous backbone Darknet19, the main difference is that it has 53 convolutional layers and usage of the residual connections.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3: Structure of the convolutional neural network Darknet53.

2.4 YOLOv4

During this iteration of development the architecture was modified with new technologies, such as Weighted-Residual-Connections, Cross-Stage-Partial-connections, Cross mini-Batch Normalization, Self-adversarial-training, Mish Activation, Mosaic data augmentation, DropBlock regularization, CIoU loss. These modifications allowed authors to reach the state-of-art benchmark on the MS COCO dataset with 43.5% AP (65.7% AP50) and approximately 65 FPS on the Tesla V100.

Also in this paper, authors introduced two terms: Bag of freebies and Bag of specials, the terms which describe different techniques that can be used during training and inference time.

- **Bag of freebies**

  It is a group of methods that only changes the training strategy or only increase the training cost. These methods are applied during the training process to get better accuracy without increasing the computational cost of the inference. To this group belong such methods:

  - data augmentation in terms of pixel-wise adjustments and in terms of changes of feature maps. For example CutOut, MixUp, DropConnect, and DropBlock. The authors proposed their own data augmentation technique Mosaic data augmentation. This technique can be considered a modification of the CutMix technique. The mosaic technique mixes 4 training images which is allowing to detect objects outside their normal context, batch normalization calculates activation statistics from 4 different images, which leads to decreasing the need for the large mini-batch.
  - Methods such as focal loss, label smoothing, and knowledge distillation, were proposed to deal with data imbalance problems in the datasets.
  - Bounding Box regression and different loss functions that are based on the IOU. For example GIoU loss, DIoU loss, and CIoU loss. That covers different characteristics of the overlapping of the ground truth and the predicted bounding boxes.

- **Bag of specials**

  It is a group of methods that increases the inference cost but significantly improve the accuracy of object detection.

- Receptive field enhancement methods, such as SPP, ASPP, and RFB.
- Attention module.

  The attention module in object detection can be divided into two groups, channel-wise attention with Squeeze-and-Excitation (SE) module and point-wise attention with Spatial Attention Module (SAM). SE can improve the power of the ResNet50 in the ImageNet classification task by 1% but it has a drawback, that method will increase the inference time by 10% of GPU. On the other hand, SAM can improve ResNet50-SE by 0.5% but it will not affect the speed of inference at all.
- Activation function.

  A good choice of activation function can make gradient propagation more efficient, and at the same time will not affect the computational cost a lot. New activation functions such as ReLU, LReLU, PReLU, hard-Swish, and Mish were proposed as a solution to the problem of vanishing gradients, which frequently occurred in classical tanh and sigmoid activation functions.

The model architecture of the YOLOv4.

- Backbone: CSPDarknet53.

  CSPDarknet53 - it is a modification of the backbone that was used in the YOLOv3. In that version of architecture was introduced the CSP layer that has the Mish activation function and two parts structure.

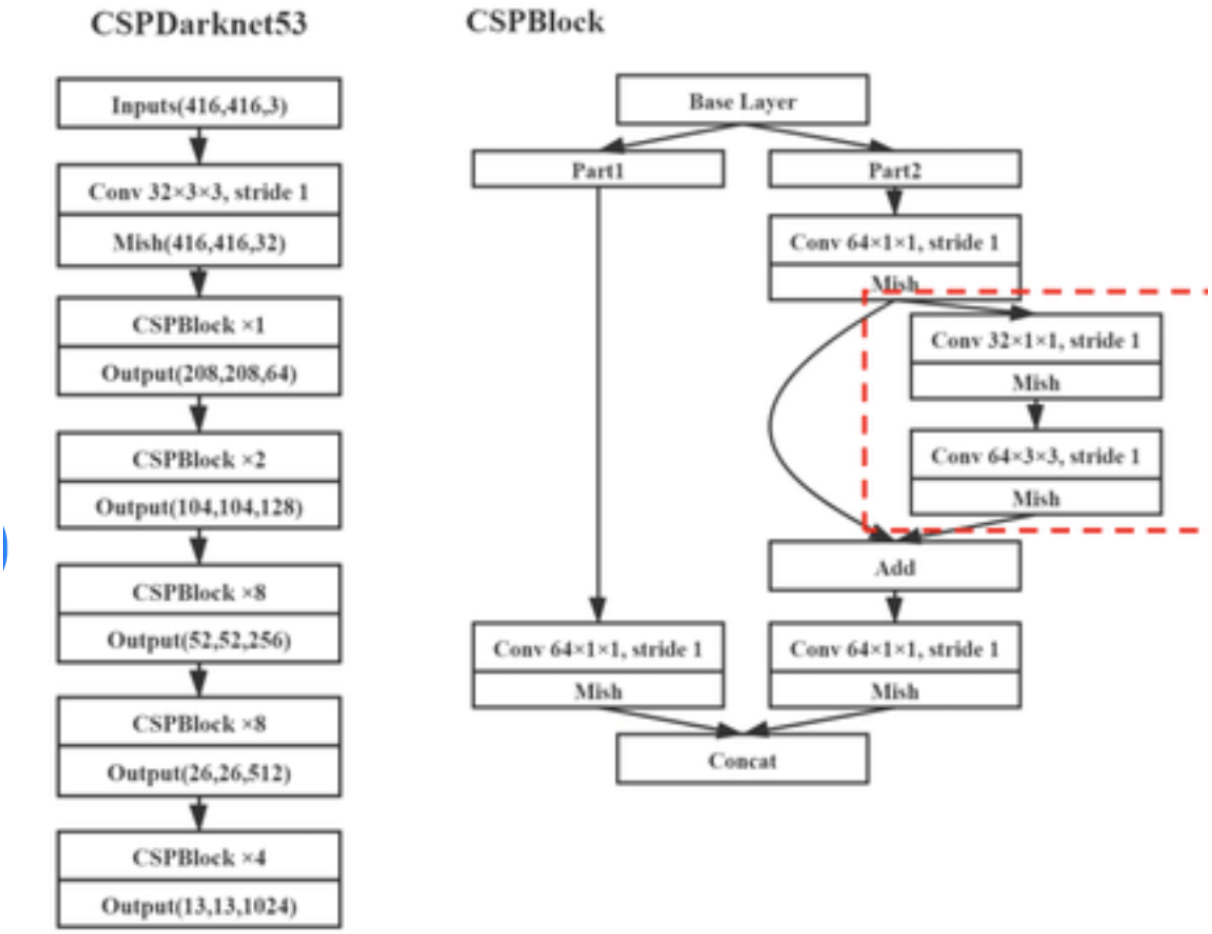

Figure 4: Architecture of the YOLOv4 model.

- Neck: SPP, PAN.
- Head: YOLOv3.

3. Experiments

In this part will be shown the effect of using different object feature extractors (backbones).

Experiments settings:

Dataset: Sampled PASCALVOC2011.

Train size = 3000 examples. **Test size**: 398 examples.

Such a decrease in the size of the dataset was caused by the lack of time and

computational capabilities.

Experiment #1 - Ground truth.
In this experiment will be used the original backbone - Darknet.
Results:

Experiment #2 - VGG16-modified.
In this experiment will be used the modification of the VGG16 network. The main change is that fully connected was changed for the object detection task.
Results:

Experiment #2 - MobileNetV1 -modifed.
In this experiment will be used the modification of the MobileNetV1 network. Change is the same as in the previous experiment.
Results:

4. Conclusions

References