Faculty of Mathematics
and Computer Science

Object detection based research methodology

Comănac Dragoș-Mihail

dragos.comanac@stud.ubbcluj.ro

Abstract

Computer vision has emerged as a key field of artificial intelligence because it aims to replicate functions of the human visual cortex. More specifically, computer vision aims to extract information from visual data such as images or videos. For example, object detection is one of the main computer vision problems in which instances of objects must be located and classified in visual data.

Deep convolutional neural networks became a standard that enabled reliable learning based data analysis, and as such, they are the key factors that turned computer vision into a very relevant field of research in our modern society. Consequently, object detection has seen the most notable advances in this era of deep learning.

The main purpose of this paper is to discuss the steps needed to create an object detection methodology using deep convolutional neural networks. We hope that this could serve as a guide for the fresh researchers that are looking to get started into this vast and complex domain.

Keywords:
Object detection; Deep learning

1. Introduction

Computer vision is a broad scientific field that deals with the extraction of meaningful information from visual data such as images or videos. As a consequence, over time, computer vision has emerged as a key field in the domain of artificial intelligence because it intends to replicate the functions of the human visual cortex. This is possible due to the continuous development of optical hardware, which nowadays can exceed the capabilities of the human eye, but probably more important are the huge quantities of data that are available to more and more people.

There are various methods that can be used in the field of computer vision, such as hand crafted features, but in this context of big data, the most relevant methods are related to machine learning, and more recently to deep learning which is better suited to visual data. Through supervised learning, deeper and deeper neural network models are now able to learn the complex patterns found in large amounts of data, thus they are a good fit for solving computer vision problems.

One such problem is object detection, which consists of locating and classifying objects in an image by producing a set of bounding boxes which contain both information about the spatial coordinates of the object, and information

about it's type or class. This problem is relevant in many domains such as automotive, logistics or even assistive technologies, thus there is an intense ongoing research into this topic. Also, it has more potential than simple classification, because the objects are localized and this forces the learning algorithm to look for the very specific patterns that describe the objects, as opposed to classification where other patterns might be learned if the data distribution is unbalanced.

Given the potential of deep neural networks in solving the problem of object detection, the purpose of this paper is to study the details of a object detection research methodology that relies on deep neural networks. We also hope that this paper could serve as a guide into this vast and complex domain.

2. Placement in the general field

By a methodology, we mean the process of applying a certain method. In our context, we focus on neural networks as a method for data analysis in the field of object detection.

The type of neural network that we are interested in falls into the category of supervised learning, which is a subfield of the more general area of machine learning. They are called deep neural networks because they have multiple hidden layers. The neural networks in the methods that are of interest for us use offline learning and are eager inductive learners, meaning that they are trained on a given dataset on which the correct output is known, in order to make predictions on a dataset on which the correct output is unknown.

It is important to note that deep neural networks have applications in various other domains such as segmentation and some of the knowledge and intuitions could be transferred between areas of interest.

The first object detection techniques heavily relied on handcrafted features. These methods were popular before 2014 and they collectively compose what is now the so called "traditional object detection period" [13]. The main innovation that took place in 2014 was that the deep neural networks reached a maturity such that they became a viable solution in general in the field of object detection. As such, the proposed object detectors since then are based on deep neural networks, hence they compose the "deep learning based detection period". The first type of solutions were quite limited, mostly due to the fact that the features were handcrafted, thus they are limited with respect to the numbers of patterns. Nevertheless, they still brought improvements that are still relevant and used by the modern detectors. Some of these innovations include regressing the bounding boxes or performing hard negative mining, which are used intensively in various object detection methodologies. Also, the sliding windows idea was developed during this time, consisting of having a windows that goes step by step over each region of the image and tries to see if there are any objects within that window. This practice as it is, is highly inefficient, but deep CNNs optimize this methodology.

2.1. One and two stage object detectors

Broadly speaking, in the context of deep neural networks, the object detection problem mainly branches out in two-stage object detection methodology which traditionally was slow, but accurate and one-stage object detection methodology which initially was less accurate and fast, but recently they became better and better, even surpassing the performance of two-stage object detectors. Also, there were significant advances brought in two-stage detectors, that made them faster and very accurate.

Initially the task of object detection was decomposed into multiple tasks, that together made a pipeline, which is difficult to train. For example, region-based object detectors such as Regions with CNN features (R-CNN) [3] and its faster variants first generate bounding boxes through selective search, then a CNN extracts features that are further used in classifying the bounding boxes. All these steps slow down performance. This approach is called two-stage object detection and it opened the way into the deep learning era of object detection.

Their counterparts, the one stage, or single-shot, object detectors achieve real-time speed with decent accuracy because their detection pipeline consists only of one CNN that processes the image and directly outputs the predictions. This approach used to have low accuracy due to the lack of large amounts of data, but recent advances have made the single-shot detectors rival the two-stage detectors in terms of accuracy, without losing speed because they benefit hardly from the end-to-end learning which allows the deep CNN to learn the various complex patterns directly from the data. Here, the You Only Look Once (YOLO) [8] method was one of the first methods to belong to the one-stage class of object detectors, hence it's success in achieving good performance, with little resources.

Also, more recently, another way of splitting the class of object detectors has emerged. Traditionally, anchors are a general way of simplifying the object detection task, and it can be applied in other tasks as well. The idea is that under ideal conditions, the machine learning algorithm can learn to predict in a specific area of an image any object with various shapes. But as it is usually the case, this proves to be difficult due to the lack of data and optimization problems. An elegant way to address this issue, is to use anchors which follow the divide et impera principle. Basically, instead of having only one predictor that focuses on various shapes, with anchors, there are several predictors that each can learn a specific shape.

3. Important steps for a object detection methodology

Over the years, a certain pattern has emerged when it comes to research in this domain that together compose a methodology. In this section we will detail the most important steps, and it is important to note that these principles can be extended to other supervised tasks.

3.1. Dataset

The first step is probably also the most important one. In order to train any machine learning algorithm, a dataset must be gathered. Its quality is crucial because it determines what the learning algorithm can actually learn.

As we have mentioned, object detection falls into the supervised class of algorithms. This means that the dataset must have a special structure, in order to satisfy the supervised algorithms. Basically, such a dataset must contain pairs of inputs and outputs from which a function that correlates them is learned.

These pairs must contain meaningful information. For example, it is a good idea to have the inputs to be samples from the same distribution in order to have some patterns. These inputs can be anything ranging from simple feature vectors to images or sequences of words and usually it is relatively easy to find lots of samples for input.

The tricky part is finding the correct output. Obviously, the outputs must be representative for the input and must describe what it should be learned. This part of gathering the outputs is especially hard because it means that each input must be processed and labeled. These labels can be quite costly, especially if expensive hardware or expert people are used in the labeling process. For instance, medical data could be such an example. Also, cheap labels can be quite costly in the end if there are a lot of inputs. And in this era of deep learning, a lot of data is required for an algorithm to learn, which is a big problem from the labeling point of view. This need for an extensive amount of good labels is one of the main pitfalls of supervised learning.

When it comes to object detection, the inputs are images, and the output represent a set of bounding boxes corresponding to the target objects in the image. They need to contain information about the class of the object and about the location of the object in the image. A common way to describe the location is to give the coordinates of the upper left and bottom right corners in the image scale.

In order to have a qualitative dataset, it is important to have labeling conventions. For instance, it is important to clearly describe what is considered to be an object of interest. Otherwise, the learning algorithm might be confused. Also, the input needs to be clear, at least for humans, because if a human can't tell what is there, it would be difficult for the learning algorithm.

Also, a common practice is to split the dataset into several parts. Firstly, a training dataset is necessary. Basically, on this, the weights of the algorithm will be updated, meaning that the patterns that describe the data are extracted from this information. Beside a training dataset, a validation dataset is necessary. No learning occurs on this data. As the name suggests, it is used to validate the quality of the algorithm. A common use is to save the models that perform good on this dataset.

This need of several datasets comes from the problem of overfitting. The idea is that if the model is too complex it will learn too well the data and it does not generalise well to unseen data. If we have a separation, we can spot overfitting by comparing the loss or another metric on the training and validation sets. If the model overfits, a continous improvement can be observed on the training set, while on validation the performance plateaus. As such, various regularization methods can be applied in order to alleviate this issue. This is a complex problem on its own.

Also, sometimes a training and validation datasets are not enough. It can happen that the by setting various hyperparameters of the learning algorithm, the validation set is also overfit. Therefore a new dataset, called test dataset is

used. The idea is that the final performance measurements are done on it because it is the only data that is not seen during optimization, and such it would provide an idea about the performance of the algorithm in a real life scenario with new data.

Before deep learning, a common choice was to have a larger proportion of the data dedicated to validation or testing such as 20-30%. This was when the datasets were smaller. But nowadays, if the dataset is large enough, this percents can be lower. For example, if the dataset contains millions of images, it may be enough to have only a fer percents dedication for validation and testing each. The intuition is that it is more important to have data on which the model can actually learn. This works best if the dataset splits are diverse enough, such that the test set is representative.

3.2. Learning algorithm

The next step is choosing the learning algorithm. Given the data description that we have mentioned before, a supervised algorithm basically learns the function from 1, where x represents the input and y the output.

$$f(x) = y \tag{1}$$

There are various supervised learning algorithms, but we are especially interested in deep convolutional neural networks. In the context of object detection, we will focus more on the one-stage paradigm.

As we have previously mentioned, the key element in one-stage methods is the fact that there is only one convolutional neural network that learns end-to-end to predict the bounding boxes from the entire image. It is a common practice to split the neural network architecture in three parts, as we have depicted in Fig. 1. Each section of this general architecture are important and deserves comprehensive studies on their own.

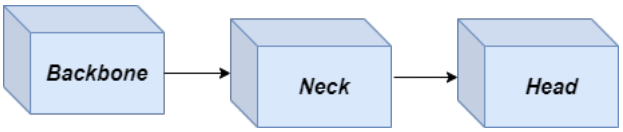

Fig. 1. General neural network architecture for one-stage object detectors

The role of the backbone is to extract feature maps from the original image, that describe for instance various shapes which can help in determining the location and class of the possible objects. Depending on the precision and speed requirements the specific design of the backbone can vary from small networks, used for speed, or large networks, used for better precision. The actual design can be custom, but if there is not enough data to accommodate the required size of the network, a common practice is to use transfer learning.

The idea is to use an already existent network architecture that is trained on a large amount of diverse data. Usually, a network used for classification is used by replacing the last layers with the neck and head architecture specific for the given problem. In this way, the extracted knowledge from the larger source dataset is stored in the network and it can be reused on the target dataset. How many layers are cut from the image classifier neural network depends on how similar are the target and source dataset. Also, regarding training, the first step is to train the whole network with the backbone frozen in order to not break the knowledge stored in its weights and the second step is to train the whole network with the backbone unfrozen, but with a low learning rate, in order to better fit the target dataset. For example a ResNet like architecture [4] can be used if high precision is needed, or a MobileNet like architecture [5, 11] is suitable if speed is of the essence.

The neck is an optional component and its main role is to be an intermediary between the backbone and the head, such that it further refines the features from the backbone in order to provide more information for the head. An example for the neck architecture is the Feature Pyramid Network [6]. The idea is that there are several feature maps extracted from the backbone that are provided to the head, in order to let the network to ”see” at several resolutions.

If the other two components are not necessarily that specific to object detection, the head is responsible for taking the output of the neck (or directly from the backbone if the neck does not exist), optionally refine them furthermore,

and finally converting them to the desired representation. Basically, it describes how the head should convert a feature maps into a representation from which bounding boxes can be extracted.

It is important that this architecture can be very well extended to other problems simply by adjusting the head.

3.3. Evaluation

To measure an object detection system performance usually the mean average precision (mAP) is the most relevant measure.

In order to see how accurate the model is, mAP is used. To compute mAP, first, the predictions are sorted descending by their confidence score. Then, the predictions are parsed one by one, and at each step, the precision and recall are computed, taking into consideration only the parsed prediction up until that point. For the recall, we consider all positives, including those that were not parsed. If we would plot these values, with the recall on the horizontal axis and the precision on the vertical axis, we would get what is called the precision-recall curve. Average precision is defined as the area under the precision-recall curve and mAP is defined as the mean of the average precisions for each class.

It is desirable to compare various methods, and in order to do that, the computation of the measurement metrics must be the same such that the comparison is relevant. Also, the data on which the comparison is done should be the same. Hence, methodologies specific to standard dataset with standard evaluations should be performed in order to see how the methods stands against other researches.

Over time, this mAP measure became the standard when it comes to measuring the quality of an object detection system.

The first standard dataset in object detection research was The Pascal Visual Object Classes (Pascal VOC) [1] and it started out as an object detection challenge. Basically, the researchers that came up with new ideas in this domain had to use this dataset in order to compare their methods with the existing literature. Additionally, in order for the comparison to be fair, the authors of the dataset also published the code for computing mAP.

As the technology advanced, researchers were able to train on larger and larger datasets, thus the Microsoft Common Objects in Context (COCO) dataset [7] became the new standard that new object detection methodologies use in order to be relevant in the field. Also, the authors provide an evaluation server which allows a better comparison between object detectors. The idea is that the labels for the test set are not publicly available, only the images can be downloaded. Therefore, the researchers that propose new object detection methods, have to upload on the evaluation server their results on the test set, and get back a detailed mAP evaluation for various thresholds and for small, medium and large objects.

3.4. Deployment

The actual deployment is usually not part of a scientific paper, because authors stop at performing experimental evaluations on the test dataset. But, we consider it relevant because the deployment is essentially the final purpose of an algorithm. As such, having in mind the actual purpose of the algorithm, the research process could be easily guided in the right direction.

For instance, embedded systems do not have that many resources, since they run on limited hardware, hence this pushes researchers to investigate algorithms that require less computations.

Also, depending on the purpose of the research methodology, this part may be the central point. Regarding deep neural networks, there is a lot of interest in quantization, which consists of compressing the weights of the neural network, in order to consume less space and perform less computations. This could be achieved by either optimizing the weights or by converting them to a representation that requires less operations, such as converting floats to integers.

4. Example of object detection methodology

In order to better understand the steps mentioned before, we detail the methodology that uses a widely used method for real-time object detection: You Only Look Once (YOLO) [8].

The main idea behind YOLO is that in order to detect multiple objects, at various locations, the image is split in a grid in which each cell is responsible for detecting objects that appear inside it. In fact, this is the representation of the ground truth labels.

4.1. Ground truth definition

The ground truth is represented as a $C \times C \times B \times (5 + N)$ tensor. For each cell in the $C \times C$ grid, there are B anchor boxes associated. Each anchor box has the following parameters: b_x and b_y represent the center of the box and they are both relative to the cell responsible for predicting the object, meaning that they are divided to the size of the cell, b_w and b_h represent the width and height, c is the probability that the anchor predicts an object and $c_1, c_2, c_3, ...c_n$ represent the N class probabilities, hence the $5 + N$ term. The actual choice for the size of the grid, the number of anchors or the number of classes is problem specific. They are important because they determine the maximum number of positives in an image. The problem of defining positives in object detection is also an important concern.

When choosing anchors there are mainly two approaches. Either they are set by hand, or they are computed through a clustering algorithm over the whole dataset. The authors for example use K-Means, with one cluster per anchor, in order to better fit the patterns of the bounding boxes in the dataset.

In the original version of the method, each cell is responsible for predicting only one bounding box, meaning that the idea of anchors is not used. As it's the case with many algorithms, anchors are only an optimization. Technically speaking, in ideal conditions, with enough data, it is enough to have one predictor per cell in order to regress any object. But as it it's often the case, there is not enough data, therefore, anchors were introduced for simplicity in the idea that it is good to split a complex problem in multiple simpler problems. The idea is that there are multiple predictors per cell that can learn various objects shapes. In this way, one predictor can specialize in tall objects and another in wide objects for instance. This also helps if there are multiple overlapped objects corresponding to one cell.

In order to create the ground truth for an image, the bounding box labels must contain the center of the object, its width and its height. The way this information is expressed can vary, but it is not important as long as the center, width and height can be computed. The next step is to find the cell in which the center of the object falls into. That cell will be responsible for detecting the object. Then the anchor that is the closest to the object in terms of shape must be chosen. This is done using Intersection Over Union (IOU).

4.2. Neural network architecture

Regarding the learning algorithm, YOLO is based on deep neural networks, that have a backbone-neck-head structure. In particular, YOLO innovates the head.

The final layer has the same structure as the ground truths. One way this can be implemented is with a 1x1 (pointwise) convolution. In order to do this, the layer before the pointwise convolution should output a feature map with a spatial size of $C \times C$. The number of channels does not matter because it is controlled by the pointwise convolution which has $B * (5 + N)$ kernels in order to output a feature map of size $C \times C \times (B * (5 + N))$ which can be further reshaped to a tensor of shape $C \times C \times B \times (5 + N)$ in order to match the shape of the ground truth.

The height, width and the coordinates of the object center are computed using the following formulas:

$$\begin{aligned} bbox_x &= \sigma(z_x) + c_x \\ bbox_y &= \sigma(z_y) + c_y \\ bbox_w &= a_w \cdot e^{z_w} \\ bbox_h &= a_h \cdot e^{z_h} \\ bbox_obj &= \sigma(z_{obj}) \end{aligned} \tag{2}$$

The network predicts the raw values z_x, z_y, z_w, z_h. The upper left corner of the predicting cell is given by c_x, c_y and are needed because the coordinates of the center are represented in the grid coordinate system. The width and height are relative to the predicting anchor, hence a_w, a_h are used in computing the width and height of the object. Another

value is predicted, that predicts the objectness z_{obj} of the bounding box, meaning that it represents the probability that there is a bounding box predicted by the anchors. Also, the softmax can be used in order to get the class probabilities, which are further multiplied with the objectness score. In order to compute the loss it is enough to leave $bbox_x$ and $bbox_y$ as they are, but they are relative to the predicting grid cell and in order to get the actual bounding box relative to the image, $bbox_x$ and $bbox_y$ need to be multiplied with the size of the grid cell in order to get them in the coordindate system of the image.

As it is the case with most object detectors, YOLO can benefit from using Non-maximum Suppression. A common problem in object detection are the overlapping bounding boxes that actually predict the same object. The role of Non-maximum Suppression is to prune away extra bounding boxes by eliminating the low confidence predictions in a certain area.

4.3. Other methodologies based on YOLO

The success of the YOLO method is given by the robustness it has shown through time. Over the years, several methodologies have been proposed that use YOLO as the key element. Each brings different optimizations over the original method, but the main aspects are still relevant. For example, the second version of YOLO [9] mainly introduces the notion of anchors and a novel multi-scale training method in order to have good predictions across images of various scales. The third version of YOLO [10] represents mostly a bundle of small improvements.

Newer object detectors such as YOLOv7 [12] and YOLOX [2] use the YOLO method in modern methodologies. The newer training techniques and optimizations resulted in competitive object detectors which achieve state of the art results. In YOLOX the authors propose an anchor free solution, and in YOLOv7 the authors continued an existing trend, that of finding methods that improve the performance without increasing the number of parameters, hence the speed loss is minimized.

4.4. Comparison between YOLO methodologies

As we have mentioned before, in order to validate a method, a methodology must be developed around a standard dataset. As such, early YOLO methods used Pascal VOC with its own computation of the mAP. In 2 we can see such a comparison.

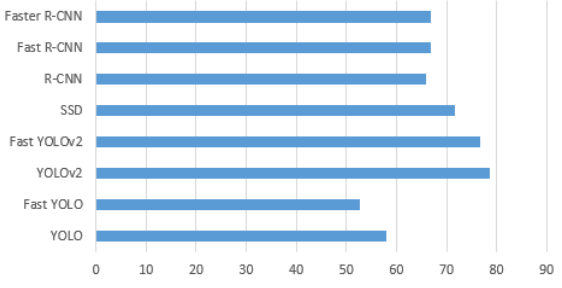

Fig. 2. Performances over the Pascal VOC dataset in terms of mAP

We also perform a comparison on the newer standard COCO dataset in Fig. 3. Here we can observe that YOLO had a constant positive evolution through time on this dataset.

5. Conclusions

In conclusion, we have detailed some of the most important steps in a object detection related research methodology such as preparing a good dataset, choosing the right learning model and evaluating it. Also, we believe that the end purpose or the deployment should be considered when designing the first parts.

Looking towards the future, we believe that this steps are generic enough to be applied not only to object detection, but to other problems as well, as the trend is to unify various computer vision tasks by combining them into a single

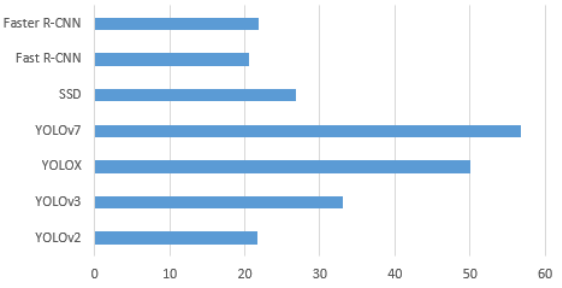

Fig. 3. Performances over the COCO dataset in terms of mAP

composed methodology. Thus, we emphasize the fact that it would be easier to develop this kind of methodology if the single tasks have the before mentioned steps similar.

References

[1] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The PASCAL Visual Object Classes (VOC) Challenge. International journal of computer vision 88, 303–338.
[2] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. YOLOX: Exceeding YOLO Series in 2021. ArXiv abs/2107.08430.
[3] Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.
[4] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.
[5] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR abs/1704.04861.
[6] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125.
[7] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context, in: European conference on computer vision, Springer. pp. 740–755.
[8] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.
[9] Redmon, J., Farhadi, A., 2017. YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 6517–6525.
[10] Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 .
[11] Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE/CVF Conference on Computer Vision and Pattern Recognition , 4510–4520.
[12] Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. ArXiv abs/2207.02696.
[13] Zou, Z., Shi, Z., Guo, Y., Ye, J., 2019. Object Detection in 20 Years: A Survey. ArXiv abs/1905.05055.