Object detection: State of the Art

Stanley Albayeros, Alejandro Zarate, Oriol Catalan, Victor Casales

Master in Computer Vision, CVC. Module 5: Visual Recognition March 2021

Abstract –In this document, we will do a quick recap of the current advances in object detection and segmentation, using the evolution of the CNN into the Mask R-CNN method.

Keywords – Object detection, object segmentation, semantic segmentation, convolutional neural network, history, overview, computer vision, neural networks.

1 Introduction

Object detection refers to the ability to identify some or all of the objects represented in an image by rough location and/or class. Object detection is usually signaled by bounding boxes (Fig.1).

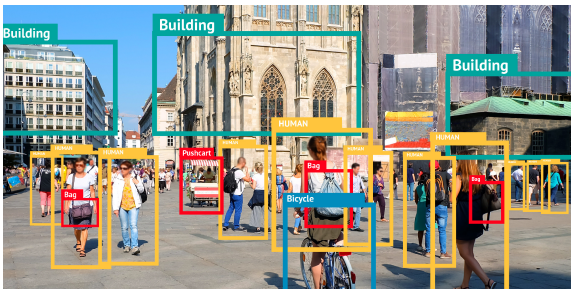

Fig. 1: Object detection

Object segmentation, also called semantic segmentation, seeks to create a per-pixel representation of the image in terms of the different objects or regions contained within(Fig. 2).

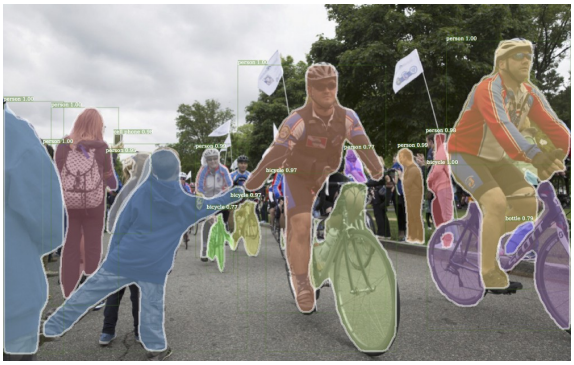

Fig. 2: Object segmentation

2 CNN: Convolutional Neural Networks

2.1 Origin

Convolutional neural networks (CNNs) were inspired by the vision processing in living organisms. In 1968, Hubel and Wiesel published a paper[1] identifying two basic cell types in the brain: simple and complex cells. According to their study, simple cells are specialized to maximize their output when they detect straight edges with particular orientations, while complex cells have a much larger field of detection, and their output is not sensitive to the exact position to the edges in their area. Hubel and Wiesel proposed the cascading model of these two cell types for use in pattern recognition.

In 1980, Fukushima[2] introduced the neocognitron, the basis of the two basic types of layers in CNNs: Convolutional layers, and Downsampling layers. The convolutional layers are the equivalent to biological simple cell types, while the downsampling layers are the equivalent to the biological complex cell types, covering large patches of the previous layers.

LeCunn et al. published in 1989[3] their mythical Lenet paper, cementing the foundations of modern computer vision as we know it.

2.2 Region based CNN: R-CNNs

We fast-forward to 2013, passing several improvements to compute power and the concepts used in CNNs. Uijlings

• Stanley Albayeros: stanley.albayeros@gmail.com
• Alejandro Zarate: alejandro.zarate@e-campus.uab.cat
• Oriol Catalán: oriol.catalan@e-campus.uab.cat
• Víctor Casales: victor.casales@e-campus.uab.cat

et al.[4] proposed a method of generating possible object locations for use in object recognition. This allowed the creation of Region-based CNNs (R-CNNs).

R-CNNs are composed of four parts:

1. Selective region search.
2. Pre-trained CNN is placed in a truncated form before the output layer.
3. Features and category of each proposed region are used to train a support vector machine (SVM) for the final object classification.
4. The features and bounding box of the proposed regions are combined and used to train a linear regression model for ground-truth prediction.

R-CNNs have the downside of being slower, even though they require the use of pre-trained CNNs. This stems from the forward computations required from the CNN to perform object detection on our proposed regions.

2.3 Fast R-CNNs

Being extremely computationally expensive, R-CNN's bottleneck is the need to extract features for each proposed region independently. Since there is a disconnect between the network dedicated to region selection and feature extraction, and these regions overlap between each other, this feature extraction process results in a very high amount of repetitive and unnecessary computations. To solve this, in 2015 Girshick[5] proposes the Fast R-CNN architecture. Compared to previous architectures, Girshick introduces Region of Interest (RoI) pooling. Girshick uses the entire image as the original CNN input for feature extraction bypassing the region proposal method. The pipeline of fast R-CNN is as follow:

1. Use the original image as the input for feature extraction, with a network trained to updatre the model parameters.
2. With N proposed regions, features detected in the same shapes are extracted from these regions of interest.
3. The CNN output and RoIs are concatenated and to summarize the features extracted from each proposed region.
4. A fully connected layer transforms the output shape to NxD, where D is determined by the model's design.
5. Softmax regression is applied during class prediction, and the shape of the fully connected layer is transformed during bounding box prediction.
6. Combining these two last changes to the shape of the layers, the class and bounding box are predict for each proposed region.

2.4 Faster R-CNN

The main issue with Fast R-CNNs is that it requires a high amount of region proposals generated in the initial selective search. This is computationally expensive and, as with the original R-CNN, results in unnecessary computations.

The Faster R-CNN architecture, proposed by Shaoqin Ren, Kaiming He, Ross Girshick and Jian Sun[6] fixes this by replacing the selective search with a completely new neural network devoted solely to region proposal: a Region Proposal Network (RPN).

The RPN is a fully convolutional network that predicts object bounding boxes and a confidence score. These bounding boxes are used as the input regions for the RoI pooling. The RPN is trained along the rest of the model, and can learn to generate high quality proposed regions, reducing the total number of regions that needs to be processed without affecting the precision of object detection negatively.

2.5 Mask R-CNN

On March 2017, Kaiming He et al.[7] publish a second paper improving their architecture even further. Kaiming et al. extend the Faster R-CNN architecture following a similiar line of thought as they did with the RPN proposal: by adding a convolutional network after the RoI align to locate objects at a pixel level within the image. This aditional CNN runs parallel with the bounding box detection and class prediction branches.

The added CNN is a feature pyramid network-styled CNN, consisting of a bottom-up pathway composed of any ConvNet/ResNet/VGG, that extracts features from raw images, a top-bottom pathway that generates a feature pyramid map, and two lateral convolution/addition operations between the corresponding levels of the two pathways. This FPN outperforms traditional ConvNets because it mantains features at various resolution scales.

2.6 Mesh R-CNN

After mask R-CNNs, in 2019 Georgia Gkioxari, Jitendra Malik and Justin Johnson[8] published a paper improving the pipeline to predict 3D shapes out of 2D images. They modify mask R-CNN in the same way the past two sections have done so: by introducing a new network to the pipeline.

Gkioxari's team introduces a mesh prediction branch that triangle meshes predicting the shapes of the detected objects in three dimentions, following a pyramidal structure. They first predict coarse representations of the features, and then use these as inputs to a graph convolution network to refine them. These meshes are then put through ShapeNet where they are validated.

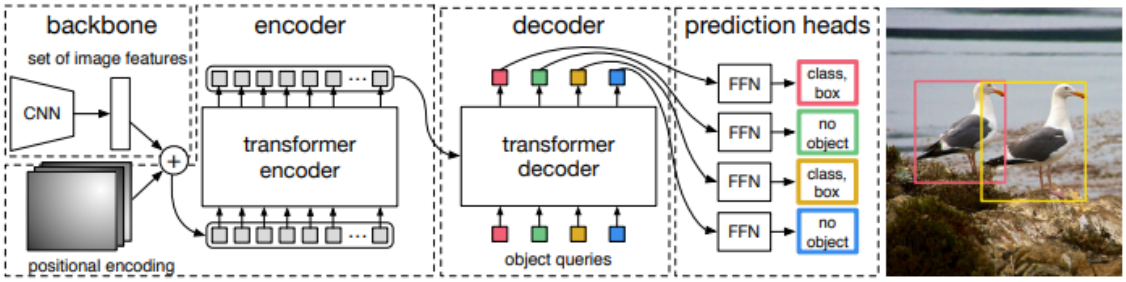

Fig. 3: End-to-End Object Detection with Transformers (DETR) Pipeline

3 End-to-End Object Detection with Transformers (DETR)

The Facebook Research team has developed an object detection model, Detection with Transformers (DETR)[9], that moves away from using R-CNN as the backbone of their pipeline, and instead utilize a transformer architecture.

Seeing how previous approaches to object detection have to deal with post-processing of the output of their components due to duplicate (or irrelevant) predictions, the DETR team seeks to simplify the object detection pipeline by shifting to a direct set prediction model, with the objective of translating the improvements that transformer networks have brought upon the Natural Language Processing(NLP) scene into the computer vision scene.

As a baseline architecture proposal, DETR seeks to match the performance of Faster R-CNN pipelines. According to the DETR paper, the new transformer-based architecture(Fig. 3) generally outperforms Faster R-CNN, except when there are many small objects, where it obtains a worse performance.

References

[1] D. H. Hubel and T. N. Wiesel. "Receptive fields and functional architecture of monkey striate cortex". In: *The Journal of Physiology* 195.1 (Mar. 1968), pp. 215–243. DOI: 10.1113/jphysiol.1968.sp008455. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1968.sp008455.

[2] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. DOI: 10.1007/bf00344251. URL: https://link.springer.com/article/10.1007%2FBF00344251.

[3] Yann Lecun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1.4 (2021), pp. 541–551. URL: https://nyuscholars.nyu.edu/en/publications/backpropagation-applied-to-handwritten-zip-code-recognition.

[4] J. R. R. Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104.2 (Apr. 2013), pp. 154–171. DOI: 10.1007/s11263-013-0620-5. URL: https://link.springer.com/article/10.1007/s11263-013-0620-5.

[5] Ross Girshick. *Fast R-CNN*. 2015. URL: https://arxiv.org/abs/1504.08083.

[6] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. URL: https://arxiv.org/abs/1506.01497.

[7] Kaiming He et al. *Mask R-CNN*. 2017. URL: https://arxiv.org/abs/1703.06870.

[8] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. *Mesh R-CNN*. 2019. URL: https://arxiv.org/abs/1906.02739.

[9] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. 2020. URL: https://arxiv.org/abs/2005.12872.