C5: Visual Recognition
Report on Object Detection and Segmentation

Team 1: Carles Pregonas Graells, Pau Vallespí Monclús, Carlos Boned Riera, Marc Pérez Sabater

Abstract— **In this report, we aim to show what have we learned in the "Visual Recognition" module. We are going to report our results, gained knowledge, insights, and difficulties that we encountered during this project.**

I. Introduction

In the beginning of this course, we started with an exploration of PyTorch, a robust deep learning framework widely acclaimed for its versatility and performance, and will compare it with Keras.

Moving forward, we'll dive into the realm of object detection, recognition, and segmentation. Armed with tools like Detectron2 [1], equipped with cutting-edge models such as Mask R-CNN[2] and Faster R-CNN[3], we'll unravel the methodologies employed to pinpoint objects within images, discern their identities, and delineate their contours with precision. After an initial inference in this models, we will perform fine tuning on the models parameters in order to find the best configuration for our data. Alongside, we'll explore the utilization of YOLO[4], an alternative framework renowned for its efficiency and simplicity in object detection tasks.

Transitioning to the realm of image retrieval, we'll peer into the mechanisms underpinning content-based image search engines. Here, sophisticated algorithms analyze image content to retrieve relevant matches from extensive databases, facilitating seamless access to visual information.

Lastly, we'll turn our attention to cross-modal retrieval—a convergence of computer vision and natural language processing. By bridging different data modalities, such as images and text, we'll explore how meaningful cross-references and information retrieval are facilitated, paving the way for enhanced understanding and utilization of multimodal data sources.

Throughout this journey, we aim to provide a comprehensive understanding of visual recognition techniques, underpinned by the utilization of sophisticated tools and methodologies. By interpreting these diverse facets, we seek to underscore the significance of visual recognition across various domains and applications.

II. Related Work

In this section, we'll delve into the narrative of object detection, exploring various perspectives such as vanilla detectors and the evolution of more sophisticated methods.

A. Vanilla Object Detectors

The transition in object detection, led by deep learning, contrasts with the pioneering era of the 1990s, where computer vision relied heavily on handcrafted features due to limited image representation. Viola-Jones Detectors in 2001 [5], [6] achieved real-time human face detection, outperforming contemporaneous algorithms through techniques like "integral image," "feature selection," and "detection cascades." Subsequent innovations, like Histogram of Oriented Gradients (HOG) [7], significantly advanced pedestrian detection. Another milestone was the introduction of Deformable Part-based Models [8], [9] in 2008, which revolutionized object detection by decomposing it into distinct parts.

In the realm of deep learning, object detectors are categorized into two main groups: *two-stage detectors* and *one-stage detectors*. The former approaches detection as a "coarse-to-fine" process, while the latter aims to complete detection in a single step.

B. Two-stage detectors

R-CNN [10] uses selective search [11]. It begins by generating a collection of 2000 object proposals (candidate boxes). These proposals are then resized to a standardized image size and passed through a pre-trained CNN model like AlexNet, trained on ImageNet, to extract features and then an SVM is used to classify them. Finally, a linear regression model is trained to generate bounding boxes for each identified object. R-CNN takes around 47 seconds for each image (inference), and the training stage is expensive and slow, as it extracts features with a CNN from 2000 regions per image.

Earlier CNN models, such as AlexNet, necessitated a fixed-size input, typically requiring images to be resized to dimensions like 224x224 pixels. A first solution to solve this problem comes from **Spatial Pyramid Pooling** [12] (**SPP**), whose goal is to get fixed-length representations for variable-size feature maps. In this case, a CNN is run just once per image to obtain a feature map, and then a (variable size) window related to the region proposals to detect the objects in the image.

This concept is further developed in **Fast R-CNN**[13], which employs RoI pooling, akin to SPP. The RoI pooling layer adjusts the region proposals to match the CNN input size. Each region is forwarded to a Fully Connected Network (FCN), where a softmax layer and a linear regression layer produce class predictions and bounding box coordinates. Fast R-CNN utilizes a combined loss function. Fast R-CNN addresses two primary challenges of R-CNN: it reduces the

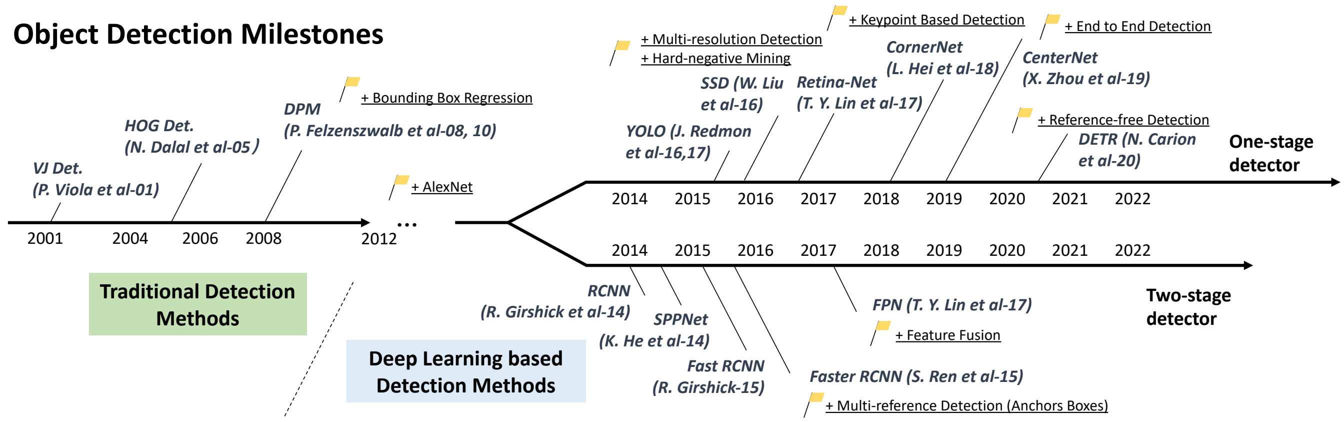

Fig. 1. Chronology of state-of-the-art object detection and instance segmentation architectures.

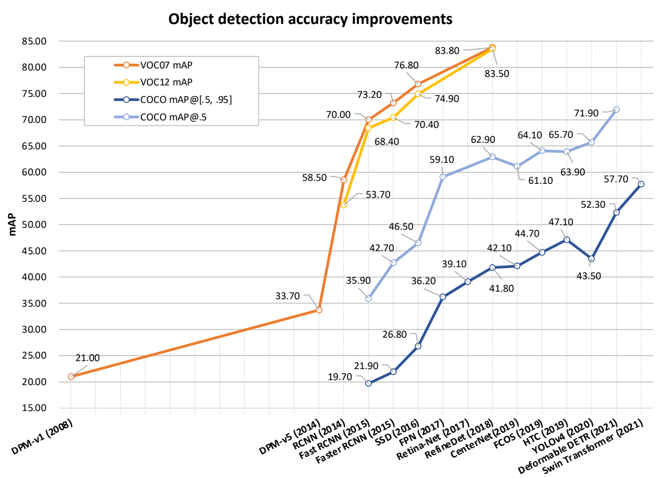

Fig. 2. Accuracy improvement in object detection across VOC07, VOC12, and MS-COCO datasets.

number of regions passed to a CNN from 2000 to one per image, and it merges feature extraction, classification, and bounding box generation into a single model. However, a new bottleneck emerges with the selective search algorithm used to locate Regions of Interest (RoIs), which is slow and time-consuming, resulting in an inference time of approximately 2 seconds per image.

To solve this problem, **Faster R-CNN** [14] introduces Region Proposal Networks (RPN): using a sliding window over the feature maps from a CNN, it generates 9 anchor boxes of different shapes and sizes at each sliding position. For each anchor, RPN predicts two things: the probability that an anchor is an object (class-agnostic), and the bounding box regressor for adjusting the anchors to better fit the object.

In Faster R-CNN, the inference time is around 0.2 seconds per image, which is much faster than R-CNN and Fast R-CNN, but is still far from real-time object detection.

C. Object instance segmentation

Moving towards segmentation, Mask R-CNN [15] builds upon Faster R-CNN by incorporating a parallel branch dedicated to predicting object masks alongside the existing branch for bounding box detection. This model applies segmentation to the Region Proposal Network (RPN) predictions, resulting in the generation of high-quality segmentation masks for each instance.

D. One-stage detectors

Instead of using region proposals, one-stage detectors make predictions by only looking into the input image once, which can result in a faster performance. One of the state-of-the-art single pass detectors is **YOLO** [16] (and its successors [17], [18], [19]) and so on upon YOLO9, which idea is to extract a feature map using a CNN called Darknet, divide it into SxS cells (YOLO uses $S = 7$), and predict one bounding box for each cell. YOLO predicts the class of that bounding box and if it is centered at that cell, and uses Non-Maximum Suppression (NMS) to reduce the number of output bounding boxes.

Another well-known one-stage detector is **SSD** [20], which makes predictions at multiple feature maps (YOLO only does it for one) using a VGG16 network [21] as a feature extractor. The idea is that each feature map, which has different local receptive fields, specializes in objects of different sizes.

The NMS method is also used at the end of the SSD model, and Hard Negative Mining (HNM) is then used to further reduce the number of predicted negative boxes.

Finally, **RetinaNet** [22] has been formed by making two improvements over existing single stage detectors: Feature Pyramid Networks (FPN) [23] and Focal Loss. FPN replaces the feature extractor of detectors like Faster R-CNN, and focal loss is introduced to handle the class imbalance problem with one-stage object detection models.

A comparison between some one-stage and two-stage detectors on Different Datasetes as COCO dataset [24] is depicted in Fig. 2.

III. Conclusions

IV. Methodology

A. Faster R-CNN & Mask R-CNN

B. Optional tasks (if applicable): description of models, methodology to generate out-of-context images

V. Experiments

A. Dataset

B. Metrics

VI. Results

VII. Conclusions

References

[1] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," *arXiv preprint arXiv:1911.02549*, 2019.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv preprint arXiv:1703.06870*, 2017.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1, pp. I–I, Ieee, 2001.

[6] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, pp. 137–154, 2004.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8, Ieee, 2008.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[11] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, pp. 154–171, 09 2013.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[13] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 06 2015.

[15] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[17] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017.

[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 04 2018.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-y. Liao, "Yolov4: Optimal speed and accuracy of object detection," 04 2020.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, "Ssd: Single shot multibox detector," vol. 9905, pp. 21–37, 10 2016.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[22] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[23] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, "Microsoft coco: Common objects in context," 05 2014.