electronics

MDPI

Article

A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit

Rafael Padilla *, Wesley L. Passos, Thadeu L. B. Dias, Sergio L. Netto and Eduardo A. B. da Silva

Electrical Engineering Program/Alberto Luiz Coimbra Institute for Post-Graduation and Research in Engineering (PEE/COPPE), PO Box 68504, Rio de Janeiro 21941-972, RJ, Brazil; wesley.passos@smt.ufrj.br (W.L.P.); thadeu.dias@smt.ufrj.br (T.L.B.D.); sergioln@smt.ufrj.br (S.L.N.); eduardo@smt.ufrj.br (E.A.B.d.S.)

* Correspondence: rafael.padilla@smt.ufrj.br

Abstract: Recent outstanding results of supervised object detection in competitions and challenges are often associated with specific metrics and datasets. The evaluation of such methods applied in different contexts have increased the demand for annotated datasets. Annotation tools represent the location and size of objects in distinct formats, leading to a lack of consensus on the representation. Such a scenario often complicates the comparison of object detection methods. This work alleviates this problem along the following lines: (i) It provides an overview of the most relevant evaluation methods used in object detection competitions, highlighting their peculiarities, differences, and advantages; (ii) it examines the most used annotation formats, showing how different implementations may influence the assessment results; and (iii) it provides a novel open-source toolkit supporting different annotation formats and 15 performance metrics, making it easy for researchers to evaluate the performance of their detection algorithms in most known datasets. In addition, this work proposes a new metric, also included in the toolkit, for evaluating object detection in videos that is based on the spatio-temporal overlap between the ground-truth and detected bounding boxes.

Keywords: object-detection metrics; precision; recall; evaluation; automatic assessment; bounding boxes

Citation: Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L., da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*, 279. https://doi.org/10.3390/electronics10030279

Academic Editor: Tomasz Trzcinski
Received: 25 December 2020
Accepted: 20 January 2021
Published: 25 January 2021

1. Introduction

The human visual system can effectively distinguish objects in different environments and contexts, even under a variety of constraints such as low illumination [1], color differences [2], and occlusions [3,4]. In addition, objects are key to the understanding of a scene's context, which lends paramount importance to the estimation of their precise location and classification. This has led computer vision researchers to explore automatic object detection for decades [5], reaching impressive results particularly in the last few years [6–9].

Object detection algorithms attempt to locate general occurrences of one or more predefined classes of objects. In a system designed to detect pedestrians, for instance, an algorithm tries to locate all pedestrians that appear within an image or a video [3,10,11]. In the identification task, however, an algorithm tries to recognize a specific instance of a given class of objects. In the pedestrian example, an identification algorithm wants to determine the identity of each pedestrian previously detected.

Initially, real-time object detection applications were limited to only one object type [12] at a time, mostly due to hardware limitations. Later on, advancements in object detection techniques led to their increasing adoption in areas that included the manufacturing industry with optical inspections [13], video surveillance [14], forensics [15,16], medical image analysis [17–19], autonomous vehicles [20], and traffic monitoring [21]. In the last decade, the use of deep neural networks (DNNs) has completely changed the landscape of the computer vision field [22]. DNNs have allowed for drastic improvements in image

classification, image segmentation, anomaly detection, optical character recognition (OCR), action recognition, image generation, and object detection [5].

The field of object detection has yielded significant improvements in both efficiency and accuracy. To validate such improvements, new techniques must be assessed against current state-of-the-art approaches, preferably over widely available datasets. However, benchmark datasets and evaluation metrics differ from work to work, often making their comparative assessment confusing and misleading. We identified two main reasons for such confusion in comparative assessments:

- There are often differences in bounding box representation formats among different detectors. Boxes could be represented, for instance by their upper-left corner coordinates (x, y) and their absolute dimensions (width, height) in pixels, or by their relative coordinates (x_{rel}, y_{rel}) and dimensions ($\text{width}_{\text{rel}}$, $\text{height}_{\text{rel}}$), with the values normalized by the image size, among others;
- Each performance assessment tool implements a set of different metrics, requiring specific formats for the ground-truth and detected bounding boxes.

Even though many tools have been developed to convert the annotated boxes from one format to another, the quality assessment of the final detections still lacks a tool compatible with different bounding box formats and multiple metrics. Our previous work [23] contributed to the research community in this direction, by presenting a tool which reads ground-truth and detected bounding boxes in a closed format and evaluates the detections using the average precision (AP) and mean average precision (mAP) metrics, as required in the PASCAL Challenge [24]. In this work that contribution is significantly expanded by incorporating 13 other metrics, as well as by supporting additional annotation formats into the developed open-source toolbox. The new evaluation tool is available at https://github.com/rafaelpadilla/review_object_detection_metrics. We believe that our work significantly simplifies the task of evaluating object detection algorithms.

This work intends to explain in detail the computation of the most popular metrics used as benchmarks by the research community, particularly in online challenges and competitions, providing their mathematical foundations and a practical example to illustrate their applicability. In order to do so, after a brief contextualization of the object-detection field in Section 2, the most common annotation formats and assessment metrics are examined in Sections 3 and 4, respectively. A numerical example is provided in Section 5 illustrating the previous concepts from a practical perspective. Popular metrics are further addressed in Section 6. In Section 7 object detection in videos is discussed from an integrated spatio-temporal point of view, and a new metric for videos is provided. Section 8 presents an open-source and freely distributed toolkit that implements all discussed concepts in a unified and validated way, as verified in Section 9. Finally, Section 10 concludes the paper by summarizing its main technical contributions.

2. An Overview of Selected Works on Object Detection

Back in the mid-50s and 60s the first attempts to recognize simple patterns in images were published [25,26]. These works identified primitive shapes and convex polygons based on contours. In the mid-80s, more complex shapes started gaining meaning, such as in [27], which described an automated process to construct a three-dimensional geometric description of an airplane.

To describe more complex objects, instead of characterizing them by their shapes, automated feature extraction methods were developed. Different methods attempted to find important feature points that when combined could describe objects broadly. Robust feature points are represented by distinctive pixels, whose neighborhood describe the same object irrespective of changes in pose, rotation, and illumination. The Harris detector [28] finds such points in the object corners based on local intensity changes. A local search algorithm using gradients was devised in [29] to solve the image registration problem, which later was expanded to a tracking algorithm [30] for identifying important points in videos.

More robust methods were able to identify characteristic pixel points and represent them as feature vectors. The so-called scale invariant feature transform (SIFT) [31], for instance, applied the difference of Gaussians in several scales coupled with histograms of gradients, yielding characteristic points with features that are robust to scale changes and rotation. Another popular feature detector and descriptor, the speed up robust features (SURF) [32], was claimed to be faster and more robust than SIFT, and uses a blob detector based on the Hessian matrix for interest point detection and wavelet responses for feature representations.

Feature-point representation methods alone are not able to perform object detection, but can help in extracting a group of keypoints that are used to represent them. In [33], the SIFT keypoints and features are used to detect humans in images, and in [34] SIFT was combined with color histograms to classify regions of interest across frames to track objects in videos. Another powerful feature extractor widely applied for object detection is the histogram of oriented gradients (HOG) [35], which is computed for several image small cells. The histograms of each cell are combined to form the object descriptor, which, associated to a classifier, can perform the object detection task [35,36].

The Viola–Jones object detection framework was described in the path-breaking work of [12]. It could detect a single class object at a rate of 15 frames per second. The proposed algorithm employed a cascade of weak classifiers to process image patches of different sizes, being able to associate bounding boxes to the target object. The Viola–Jones method was first applied to face detection and required extensive training to automatically select a group of Haar-features to represent the target object, thus detecting one class of objects at a time. This framework has been extended to detect other object classes such as pedestrians [10,11] and cars [37].

More recently, with the growth and popularization of deep learning in computer vision problems [6,7,38–40], object detection algorithms have started to develop from a new perspective [41,42]. The traditional feature extraction [31,32,35] phase is performed by convolutional neural networks (CNNs), which are dominating computer vision research in many fields. Due to their spatial invariance, convolutions perform feature extraction spatially and can be combined into layers to produce the desired feature maps. The network end is usually composed of fully connected (FC) layers that can perform classification and regression tasks. The output is then compared to a desired result and the network parameters are adjusted to minimize a given loss function. The advantage of using DNNs in object detection tasks is the fact that their architectures can extract features and predict bounding boxes in the same pipeline, allowing efficient end-to-end training. The more layers a network has, the more complex features it is able to extract, but the more parameters it needs to learn, demanding more computer processing power and data.

When it is not feasible to acquire more real data, data augmentation techniques are used to generate artificial but realistic data. Color and geometric operations and changes inside the target object area are the main actions performed by data augmentation methods for object detection tasks [43]. The work in [44] applied generative adversarial networks (GANs) to increase by 10 times the amount of medical chest images to detect patients with COVID-19. In [45], the number of images was increased by applying filters in astronomy images so as to improve the performance of galaxy detectors.

The CNN-based object detectors may be cataloged as single-shot or region-based detectors, also known as one- or two-stage detectors, respectively. The single-shot detectors work by splitting the images into a grid of cells. For each cell, they make bounding-box guesses of different scales and aspect ratios. This type of detector prioritizes speed rather than accuracy, aiming to predict both bounding box and class simultaneously. Overfeat [46] was one of the first single-shot detectors, followed by the single shot multiBox detector (SSD) [47], and all versions of you only look once (YOLO) [9,48–51]. The region-based detectors perform the detection in two steps. First, they generate a sparse set of region proposals in the image where the objects are supposed to be. The second stage classifies each object proposal and refines its estimated position. The region-based convolutional

neural network (R-CNN) [52] was a pioneer employing CNNs in this last stage, achieving significant gains in accuracy. Later works such as Fast R-CNN [53], Faster R-CNN [8], and region-based fully convolutional networks (R-FCN) [54] suggest changes in R-CNN to improve its speed. The aforementioned detectors have some heuristic and hand-crafted steps such as region feature extraction or non-maximum suppression to remove duplicate detections. In this context, graph neural networks (GNNs) are employed to compute region of interest features in a more efficient way and process the objects simultaneously by modeling them according to their appearance feature and geometry [55,56].

Hybrid solutions combining different approaches have been proposed lately and have proved to be more robust in various object-detection applications. The work in [57] proposes a hybrid solution involving a genetic algorithm and CNNs to classify small objects (structures) presented in microscopy images. Feature descriptors coupled with a cuckoo search algorithm were applied by the authors of [58] to detect vessels in a marine environment using synthetic aperture radar (SAR) images. This approach was compared to genetic algorithms and neural network models individually, improving precision to nearly 96.2%. Vision-based autonomous vehicles can also benefit from hybrid models as shown in [59], where a system integrating different approaches was developed to detect and identify pedestrians and to predict their movements. In the context of detecting objects using depth information, the work in [60] proposes a hybrid attention neural network that incorporates depth and high-level RGB features to produce an attention map to remove background information.

Other works aim to detect the most important region of interest and segment relevant objects using salient object detectors. The work in [61] proposes a pipeline to separate an input image into a pair of images using content-preserving transforms. Then, each resulting image is passed by an interweaved convolutional neural network, which extracts complementary information of the image pairs and fuses them into the final salient map.

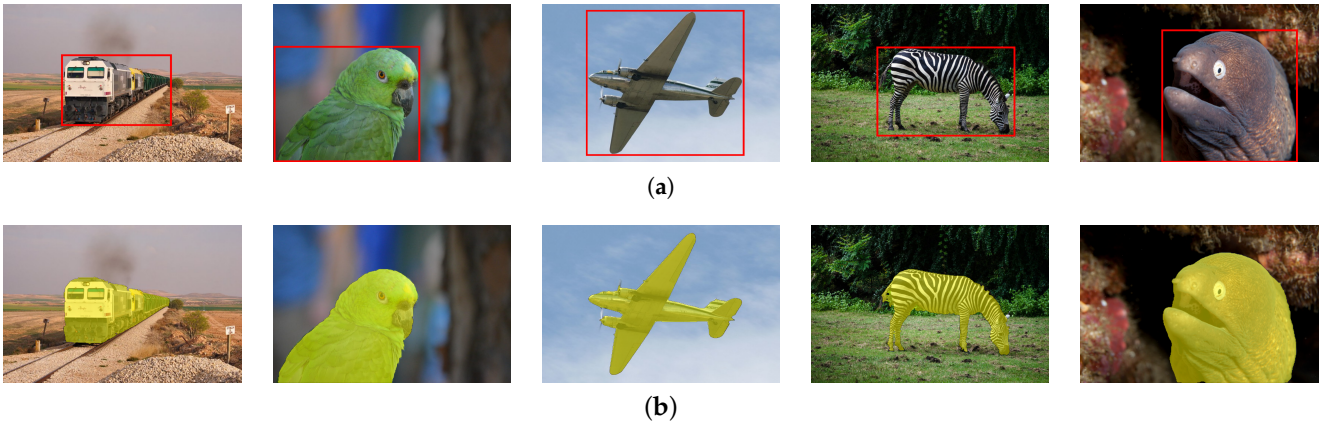
As medical images are acquired with special equipment, they form a very specific type of image [17]. To detect lesions, organs, and other structures of interest can be crucial for a precise diagnostic. However, most object detection systems are designed for general applications and usually do not perform well in medical images without adaptations [62]. Detecting anomalies such as glaucoma, breast, and lung lesions, for instance, have been explored from the medical object-detection perspective in [63,64]. In the medical field, training and testing data usually have significant differences due to data scarcity and privacy. In order to address this issue, a domain adaptation framework, referred to as clustering CNNs (CLU-CNNs) [65], has been proposed to improve the generalization capability without specific domain training.

With new object detection methods being constantly released, it is highly desirable that a consensual evaluation procedure is established. To do so, the most common bounding box formats used by public datasets and competitions are revised in the next section.

3. Bounding Box Formats

Given the large, ever growing number of object detectors based on supervised methods in different areas, specific datasets have been built to train these systems. Popular competitions such as the common objects in context (COCO) [66], PASCAL visual object classes (VOC) [67], and Open Images Dataset [68] offer annotated datasets so that participants can train and evaluate their models before submission. Apart from using available datasets, collecting data for a object detection task can be quite challenging, as labeling images and videos is often a manual and quite demanding exercise. The medical field provides a good example of this fact: Developing a database of X-ray, electroencephalography (EEG), magnetoencephalography (MEG), or electrocardiography (ECG) images involves not only high costs for capturing such signals, but also requires expert knowledge to interpret and annotate them.

To ease the object annotation process in images and videos, many tools have been developed to annotate different datasets. Such tools basically offer the same features, such as bounding-box annotations and polygon-like silhouettes, as shown in Figure 1.



(**a**)

(**b**)

Figure 1. Two different types of annotated images from OpenImage [68]: (**a**) Bounding box annotations; (**b**) Silhouette annotations (in yellowish-green), also referred to as segmentation and pixel-level annotations.

A vast amount of annotation tools are freely available. Table 1 lists the most popular ones with their respective bounding box output formats.

Table 1. Popular free annotation tools and their supported output formats.

Annotation Tool	Annotation Types	Output Formats
LabelMe [69]	Bounding boxes and polygons	LabelMe, but provides conversion to COCO and PASCAL VOC
LabelIMG [70]	Bounding boxes	PASCAL VOC and YOLO
Microsoft VoTT [71]	Bounding boxes and polygons	PASCAL VOC, TFRecords, specific CSV, Azure Custom Vision Service, Microsoft Cognitive Toolkit (CNTK), VoTT
Computer Vision Annotation Tool (CVAT) [72]	Bounding boxes and polygons	COCO, CVAT, LabelMe, PASCAL VOC, TFRecord, YOLO, and others
VGG Image Annotation Tool (VIA) [73]	Bounding boxes and polygons	COCO and specific CSV and JSON

Some datasets introduced new formats to represent their annotations, which are usually named after the datasets themselves. The PASCAL VOC dataset [67] established the PASCAL VOC XML format and the COCO dataset [66] represents their annotations in the COCO format, embodied in a JSON file. Annotation tools also brought further formats. For example, CVAT [72], a popular annotation tool, outputs bounding boxes in multiple formats, including its own specific XML-based one, named a CVAT format. The most popular bounding box formats shown in Table 1 are described in more detail. Note that whenever we refer to absolute coordinates, we mean coordinates that are expressed on the image coordinate frame, as opposed to coordinates that are normalized by the image width or image height.

1. PASCAL VOC: It consists of one XML file for each image containing none, one or multiple bounding boxes. The upper-left and bottom-right pixel coordinates are absolute. Each bounding box also contains a tag representing the class of the object. Extra information about the labeled object can be provided, such as whether, the object extends beyond the bounding box or it is partially occluded. The annotations in the ImageNet [74] and PASCAL VOC [67] datasets are provided using the PASCAL VOC format;

2. COCO: It is represented by a single JSON file containing all bounding boxes of a given dataset. The classes of the objects are listed separately in the *categories* tag and identified by an *id*. The image file corresponding to an annotation is also indicated in a separate element (*images*) that contains its file name and is referenced by an *id*. The bounding boxes and their object classes are listed in a different element (*annotations*), with their top-left (x, y) coordinates being absolute, and with explicit values of width and height;
3. LabelMe: The bounding-box annotations in this format are inserted in a single JSON file for each image, containing a list of boxes represented by their absolute upper-left and bottom-right coordinates. Besides the class of the object, this format also contains the image data encoded in base64 type, thus making the LabelMe format to consume more storage space than others;
4. YOLO: One TXT file per image is used in this representation. Each line of the file contains the class id and the bounding box coordinates. An extra file is needed to map the class id to the class name. The bounding box coordinates are not absolute, being represented by the format $\left(\frac{x_{\text{center}}}{\text{image width}}, \frac{y_{\text{center}}}{\text{image height}}, \frac{\text{box width}}{\text{image width}}, \frac{\text{height}}{\text{image height}}\right)$. The advantage of representing the boxes in this format is that, if the image dimensions are scaled, the bounding box coordinates do not change, and thus the annotation file does not have to be altered. This type of format is the one preferred by those who annotate images in one resolution and need to scale their dimensions to fulfill the input shape requirement of a specific CNN. The YOLO object detector needs bounding boxes in this format to execute training;
5. VoTT: This representation of the bounding boxes coordinates and object class is made in a JSON file (one file per image) and the coordinates are expressed as the width, height and upper-left (x, y) pixel position in absolute coordinates. The Visual Object Tagging Tool (VoTT) produces annotations in this format;
6. CVAT: It consists of a unique XML file with all bounding boxes in the dataset represented by the upper-left and bottom-right pixel absolute coordinates. This format has been created with the CVAT annotation tool;
7. TFRecord: This is a serialized representation of the whole dataset containing all images and annotations in a single file. This format is recognized by the Tensorflow library [75];
8. Tensorflow Object Detection: This is a CSV file containing all labeled bounding boxes of the dataset. The bounding box format is represented by the upper-left and bottom-right pixel absolute coordinates. This is also a widely used format employed by the Tensorflow library;
9. Open Images Dataset: This format is associated with the Open Images Dataset [68] to annotate its ground-truth bounding boxes. All annotations are written in a unique CSV file listing the name of the images and labels, as well as upper-left and bottom-right absolute coordinates of the bounding boxes. Extra information about the labeled object is conveyed by other tags such as, for example, *IsOcclude*, *IsGroupOf*, and *IsTruncated*.

As each dataset is annotated using a specific format, works tend to employ the evaluation tools provided along with the datasets to assess their performance. Therefore, their results are dependent on the specific metric implementation associated with the used dataset. For example, the PASCAL VOC dataset employs the PASCAL VOC annotation format, which provides a MATLAB code implementing the metrics AP and mAP (intersection over union (IOU)=.50). This tends to inhibit the use of other metrics to report results obtained for this particular dataset. Table 2 lists popular object detection methods along with the datasets and the 14 different metrics used to report their results, namely: AP@[.5:.05:.95], AP@.50, AP@.75, AP_S, AP_M, AP_L, AR_1, AR_{10}, AR_{100}, AR_S, AR_M, AR_L, mAP (IOU=.50), and AP.

As the evaluation metrics are directly associated with a given annotation format, almost all works report their results only for the metrics implemented for the benchmarking dataset. For example, mAP (IOU=.50) is reported when the PASCAL VOC dataset is used,

while AP@[.5:.05:.95] is applied to report results on the COCO dataset. If a work uses the COCO dataset to train a model and wants to evaluate their results with the PASCAL VOC tool, it will be necessary to convert the ground-truth COCO JSON format to the PASCAL VOC XML format. This scenario discourages the use of such cross-dataset assessments, which have become quite rare in the object detection literature.

Table 2. Popular object detection methods along with the datasets and metrics used to report their results.

Method	Benchmark Dataset	Metrics
CornerNet [76]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L
EfficientDet [77]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75
Fast R-CNN [53]	PASCAL VOC 2007, 2010, 2012	AP; mAP (IOU=.50)
Faster R-CNN [8]	PASCAL VOC 2007, 2012	AP; mAP (IOU=.50)
Faster R-CNN [8]	COCO	AP@[.5:.05:.95]; AP@.50
R-CNN [52]	PASCAL VOC 2007, 2010, 2012	AP; mAP (IOU=.50)
RFB Net [78]	PASCAL VOC 2007	mAP (IOU=.50)
RFB Net [78]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L
RefineDet [79]	PASCAL VOC 2007, 2012	mAP (IOU=.50)
RefineDet [79]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L
RetinaNet [80]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L
R-FCN [54]	PASCAL VOC 2007, 2012	mAP (IOU=.50)
R-FCN [54]	COCO	AP@[.5:.05:.95];AP@.50; AP_S; AP_M; AP_L
SSD [47]	PASCAL VOC 2007, 2012	mAP (IOU=.50)
SSD [47]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L; AR_1; AR_{10}; AR_{100}; AR_S; AR_M; AR_L
SSD [47]	ImageNet	mAP (IOU=.50)
Yolo v1 [48]	PASCAL VOC 2007, 2012; Picasso; People-Art	AP; mAP (IOU=.50)
Yolo v2 [49]	PASCAL VOC 2007, 2012	AP; mAP (IOU=.50)
Yolo v2 [49]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L; AR_1; AR_{10}; AR_{100}; AR_S; AR_M; AR_L
Yolo v3 [50]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L; AR_1; AR_{10}; AR_{100}; AR_S; AR_M; AR_L
Yolo v4 [51]	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; AP_S; AP_M; AP_L
Yolo v5 [9]	COCO	AP@[.5:.05:.95]; AP@.50

An example of confusions that may arise in such a scenario is given by the fact that some works affirm that the metrics AP@.50 and mAP (IOU=.50) are the same [54], which may not always be true. The origins of such misunderstandings are the differences in how each tool computes the corresponding metrics. The next section deals with this problem by detailing the implementations of the several object detection metrics and pointing out their differences.

4. Performance Metrics

Challenges and online competitions have pushed forward the frontier of the object detection field, improving results for specific datasets in every new edition. To validate the submitted results, each competition applies a specific metric to rank the submitted detections. These assessment criteria have also been used by the research community to report and compare object detection methods using different datasets as illustrated in Table 2. Among the popular metrics to report the results, this section will cover those used by the most popular competitions, namely Open Images RVC [81], COCO Detection Challenge [82], VOC Challenge [24], Datalab Cup [83], Google AI Open Images challenge [84], Lyft 3D Object Detection for Autonomous Vehicles [85], and City Intelligence Hackathon [86]. Object detectors aim to predict the location of objects of a given class in an image or video with a high confidence. They do so by placing bounding boxes to identify the positions of the objects. Therefore, a detection is represented by a set of three attributes: The object class, the corresponding bounding box, and the confidence score, usually given

by a value between 0 and 1 showing how confident the detector is about that prediction. The assessment is done based on:

- A set of ground-truth bounding boxes representing the rectangular areas of an image containing objects of the class to be detected, and
- a set of detections predicted by a model, each one consisting of a bounding box, a class, and a confidence value.

Detection evaluation metrics are used to quantify the performance of detection algorithms in different areas and fields [87,88]. In the case of object detection, the employed evaluation metrics measure how close the detected bounding boxes are to the ground-truth bounding boxes. This measurement is done independently for each object class, by assessing the amount of overlap of the predicted and ground-truth areas.

Consider a target object to be detected represented by a ground-truth bounding box B_{gt} and the detected area represented by a predicted bounding box B_p. Without taking into account a confidence level, a perfect match is considered when the area and location of the predicted and ground-truth boxes are the same. These two conditions are assessed by the intersection over union (IOU), a measurement based on the Jaccard Index, a coefficient of similarity for two sets of data [89]. In the object detection scope, the IOU is equal to the area of the overlap (intersection) between the predicted bounding box B_p and the ground-truth bounding box B_{gt} divided by the area of their union, that is:

$$J(B_p, B_{gt}) = \text{IOU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})}, \tag{1}$$

as illustrated in Figure 2.

$$\text{IOU} = \frac{\text{area of overlap}}{\text{area of union}} =$$

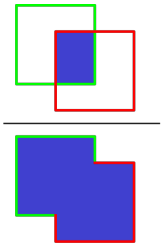


Figure 2. Illustration of the intersection over union (IOU).

A perfect match occurs when IOU $= 1$ and, if both bounding boxes do not intercept each other, IOU $= 0$. The closer to 1 the IOU gets, the better the detection is considered. As object detectors also perform the classification of each bounding box, only ground-truth and detected boxes of the same class are comparable through the IOU.

By setting an IOU threshold, a metric can be more or less restrictive on considering detections as correct or incorrect. An IOU threshold closer to 1 is more restrictive as it requires almost-perfect detections, while an IOU threshold closer to, but different than 0 is more flexible, considering as detections even small overlaps between B_p and B_{gt}. IOU values are usually expressed in percentages, and the most used threshold values are 50% and 75%. In Sections 4.1, 4.2, and 4.4 the IOU is used to define the metrics that are most relevant to object detection.

4.1. Precision and Recall

Let us consider a detector that assumes that every possible rectangular region of the image contains a target object (this would be done by placing bounding boxes of all possible sizes centered in every image pixel). If there is one object to be detected, the detector would correctly find it by one of the many predicted bounding boxes. That is not an efficient way to detect objects, as many wrong predictions are made as well. Conversely, a detector which never generates any bounding box, will never have a miss-detection. These extreme

examples highlight two important concepts, referred as precision and recall, are further explained below.

Precision is the ability of a model to identify only relevant objects. It is the percentage of correct positive predictions. Recall is the ability of a model to find all relevant cases (all ground-truth bounding boxes). It is the percentage of correct positive predictions among all given ground truths. To calculate the precision and recall values, each detected bounding box must first be classified as:

- True positive (TP): A correct detection of a ground-truth bounding box;
- False positive (FP): An incorrect detection of a non-existing object or a misplaced detection of an existing object;
- False negative (FN): An undetected ground-truth bounding box.

Assuming there is a dataset with G ground-truths and a model that outputs N detections, of which S are correct ($S \leq G$), the concepts of precision and recall can be formally expressed as:

$$Pr = \frac{\sum_{n=1}^{S} \mathrm{TP}_n}{\sum_{n=1}^{S} \mathrm{TP}_n + \sum_{n=1}^{N-S} \mathrm{FP}_n} = \frac{\sum_{n=1}^{S} \mathrm{TP}_n}{\text{all detections}}, \tag{2}$$

$$Rc = \frac{\sum_{n=1}^{S} \mathrm{TP}_n}{\sum_{n=1}^{S} \mathrm{TP}_n + \sum_{n=1}^{G-S} \mathrm{FN}_n} = \frac{\sum_{n=1}^{S} \mathrm{TP}_n}{\text{all ground truths}}. \tag{3}$$

4.2. Average Precision

As discussed above, the output of an object detector is characterized by a bounding box, a class, and a confidence interval. The confidence level can be taken into account in the precision and recall calculations by considering as positive detections only those whose confidence is larger than a confidence threshold τ. The detections whose confidence level is smaller than τ are considered as negatives. By doing so, one may rewrite Equations (2) and (3) to consider this dependence on the confidence threshold τ as:

$$Pr(\tau) = \frac{\sum_{n=1}^{S} \mathrm{TP}_n(\tau)}{\sum_{n=1}^{S} \mathrm{TP}_n(\tau) + \sum_{n=1}^{N-S} \mathrm{FP}_n(\tau)} = \frac{\sum_{n=1}^{S} \mathrm{TP}_n(\tau)}{\text{all detections}(\tau)}, \tag{4}$$

$$Rc(\tau) = \frac{\sum_{n=1}^{S} \mathrm{TP}_n(\tau)}{\sum_{n=1}^{S} \mathrm{TP}_n(\tau) + \sum_{n=1}^{G-S} \mathrm{FN}_n(\tau)} = \frac{\sum_{n=1}^{S} \mathrm{TP}_n(\tau)}{\text{all ground truths}}. \tag{5}$$

Both $\mathrm{TP}(\tau)$ and $\mathrm{FP}(\tau)$ are decreasing functions of τ, as a larger τ reduces the number of positive detections. Conversely, $\mathrm{FN}(\tau)$ is an increasing function of τ, since less positive detections imply a larger number of negative detections. In addition, $\sum \mathrm{TP}(\tau) + \sum \mathrm{FN}(\tau)$ does not depend on τ and is a constant equal to the number of all the ground truths. Therefore, from Equation (5), the recall $Rc(\tau)$ is a decreasing function of τ. On the other hand, nothing can be said a priori about the precision $Pr(\tau)$, since both the numerator and denominator of Equation (4) are decreasing functions of τ, and indeed the graph of

$Pr(\tau) \times Rc(\tau)$ tends to exhibit a zig-zag behavior in practical cases, as later illustrated in Section 5.

In practice, a good object detector should find all ground-truth objects (FN $= 0 \equiv$ high recall), while identifying only relevant objects (FP $= 0 \equiv$ high precision). Therefore, a particular object detector can be considered good if, when the confidence threshold decreases, its precision remains high as its recall increases. Hence, a large area under the curve (AUC) tends to indicate both high precision and high recall. Unfortunately, in practical cases, the precision $\times$ recall plot is often not monotonic, being zigzag-like instead, which poses challenges to an accurate measurement of its AUC.

The average precision (AP) is a metric based on the area under a $Pr \times Rc$ curve that has been pre-processed to eliminate the zig-zag behavior. It summarizes this precision-recall trade-off dictated by confidence levels of the predicted bounding boxes.

To compute the AP, one starts by ordering the K different confidence values $\tau(k)$ output by the object detector as:

$$\tau(k), \quad k = 1, 2, \ldots, K \quad \text{such that} \quad \tau(i) > \tau(j) \ \text{ for } \ i > j. \tag{6}$$

Since the Rc values also have a one-to-one, monotonic correspondence with τ, which has a one-to-one, monotonic, correspondence with the index k, then the $Pr \times Rc$ curve is not continuous but sampled at the discrete points $Rc(\tau(k))$, leading to the set of pairs $(Pr(\tau(k), Rc(\tau(k))$ indexed by k.

Now one defines an ordered set of reference recall values $R_r(n)$,

$$R_r(n), \quad n = 1, 2, \ldots, N \quad \text{such that} \quad R_r(m) < R_r(n) \ \text{ for } \ m > n. \tag{7}$$

The AP is computed using the two ordered sets in Equations (6) and (7). But before computing AP, the precision $\times$ recall pairs have to be interpolated such that the resulting precision $\times$ recall curve is monotonic. The resulting interpolated curve is defined by a continuous function $Pr_{\text{interp}}(R)$, where R is a real value contained in the interval $[0, 1]$, defined as:

$$Pr_{\text{interp}}(R) = \max_{k|Rc(\tau(k)) \geq R} \{Pr(\tau(k))\}, \tag{8}$$

where $\tau(k)$ is defined in Equation (6) and $Rc(\tau(k))$ is the recall value for the confidence $\tau(k)$, computed according to Equation (5). The precision value interpolated at recall R corresponds to the maximum precision $Pr_{\text{interp}}(k)$ whose corresponding recall value is greater than or equal to R. Note that an interpolation using a polynomial fitting would not be convenient in this case, since a polynomial interpolation cannot guarantee that the resulting interpolated curve is monotonic.

Now one is ready to compute AP by sampling $Pr_{\text{interp}}(R)$ at the N reference recall values R_r defined in Equation (7). The AP is the area under the $Pr \times Rc$ curve calculated by a Riemann integral of $Pr_{\text{interp}}(R)$ using the K recall values from the set $R_r(k)$ in Equation (7) as sampling points, that is,

$$\text{AP} = \sum_{k=0}^{K} (R_r(k) - R_r(k+1)) Pr_{\text{interp}}(R_r(k)), \tag{9}$$

where $Pr_{\text{interp}}(R)$ is defined in Equation (8) and $Rr(k)$ is given by Equation (12), with $R_r(0) = 1$ and $R_r(K+1) = 0$.

There are basically two approaches to compute this Riemann integral: The N-point interpolation and the all-point interpolation, as detailed below.

4.2.1. N-Point Interpolation

In the N-point interpolation, the set of reference recall values $R_r(n)$ for the computation of the Riemann integral in Equation (9) are equally spaced in the interval $[0, 1]$, that is,

$$R_r(n) = \frac{N-n}{N-1}, \quad n = 1, 2, \ldots, N. \tag{10}$$

and thus the expression for AP becomes:

$$\text{AP} = \frac{1}{N} \sum_{n=1}^{N} Pr_{\text{interp}}(R_r(n)). \tag{11}$$

Actually the N-point interpolation as defined by Equation (11) computes an AP value which is equal to the value computed by the Riemann integral in Equation (9) multiplied by $\frac{N-1}{N}$.

Popular applications of this interpolation method use $N = 101$ as in the competition [82] and $N = 11$ as initially adopted by the competition [24], which was later changed to the all-point interpolation method.

4.2.2. All-Point Interpolation

For the computation of AP using the so-called all-point interpolation, here referred to as AP_{all}, as the set values $R_r(n)$ used to compute the Riemann integral in Equation (9) corresponds exactly to the set of recall values computed considering all K confidence levels $\tau(k)$ in Equation (6), with the confidences $\tau(0) = 0$ and $\tau(K+1) = 1$ added so that the points $R_r(0) = 1$ and $R_r(K+1) = 0$ are considered in Equation (9). More precisely,

$$\begin{aligned} R_r(0) &= 1, \\ R_r(k) &= Rc(\tau(k)), \quad k = 1, 2, \ldots, K, \\ R_r(K+1) &= 0. \end{aligned} \tag{12}$$

where $Rc(\tau(k))$ is given by Equation (5) with $Rc(\tau(0)) = 1$ and $Rc(\tau(K+1)) = 0$.

Using this definition of $R_r(k)$ in Equation (12), AP_{all} is computed using Equation (9). In the all-point interpolation, instead of using the precision observed at only a few points, the AP is obtained by interpolating the precision at each recall level. The Pascal Challenge [24] adopts the all-point interpolation method to compute the average precision.

4.3. Mean Average Precision

Regardless of the interpolation method, AP is obtained individually for each class. In large datasets with many classes, it is useful to have a unique metric that is able to represent the exactness of the detections among all classes. For such cases, the mean average precision (mAP) is computed, which is simply the average AP over all classes [8,47], that is,

$$\text{mAP} = \frac{1}{C} \sum_{i=1}^{C} \text{AP}_i, \tag{13}$$

where AP_i is the AP value for the i-th class and C is the total number of classes being evaluated.

4.4. Average Recall

The average recall (AR) [90] is another evaluation metric used to measure the assertiveness of object detectors for a given class. Unlike the average precision, the confidences of the estimated detections are not taken into account in AR computation. This turns all detections into positive ones, which is equivalent to setting the confidence threshold as $\tau = 0$ in Equations (4) and (5).

The AR metric makes an evaluation at a large range of IOU thresholds, by taking into account all recall values obtained for IOU thresholds in the interval $[0.5, 1]$. An IOU of 0.5 can be interpreted as a rough localization of an object and is the least acceptable IOU by most of the metrics, and an IOU equal to 1 is equivalent to the perfect location of the detected object. Therefore, by averaging recall values in the interval $[0.5, 1]$, the model is evaluated on the condition of the object location being considerably accurate.

Let o be the IOU overlap between a ground truth and a detected bounding box as computed by Equation (1), and $Rc_{IOU}(o)$ a function that retrieves the recall for a given IOU o. The AR is defined as twice the area under the $Rc_{IOU}(o) \times o$ curve for the IOU interval $[0.5, 1]$, that is,

$$\text{AR} = 2 \int_{0.5}^{1} Rc_{IOU}(o)\, do. \tag{14}$$

The authors in [90] also give a straightforward equation for the computation of the above integral from the discrete sample set, as twice the average of the excess IOU for all the ground-truths, that is,

$$\text{AR} = \frac{2}{G} \sum_{i=1}^{G} \max(\text{IOU}_i - 0.5, 0), \tag{15}$$

where IOU_i denotes the best IOU obtained for a given ground truth i and G is the total number of ground-truths.

Interestingly, COCO also reports the AR, although its definition does not match exactly that in Equation (15). Instead, what is reported as the COCO AR is the average of the maximum obtained recall across several IOU thresholds. To do so one first defines a set of O IOU thresholds:

$$t(o), \quad o = 1, 2, \ldots, O. \tag{16}$$

Then, letting $Pr_{t(o)}(\tau(k)), Rc_{t(o)}(\tau(k))$ be the precision $\times$ recall points for a confidence $\tau(k)$, given the IOU threshold $t(o)$, the COCO AR is computed as:

$$\text{AR} = \frac{1}{O} \sum_{o=1}^{O} \max_{k | Pr_{t(o)}(\tau(k)) > 0} \{Rc_{t(o)}(\tau(k))\}, \tag{17}$$

that is, the average of the largest recall values such that the precision is greater than zero for each IOU threshold, and $\tau(k)$ as defined in Equation (6). Effectively, this yields a coarse approximation of the original integral in Equation (14), provided that the IOU threshold set $t(o)$ covers an adequate range of overlaps.

4.5. Mean Average Recall

As the AR is calculated individually for each class, similarly to what is done to compute mAP, a unique AR value can be obtained considering the mean AR among all classes, that is:

$$\text{mAR} = \frac{1}{C} \sum_{i=1}^{C} \text{AR}_i. \tag{18}$$

In the sequel, a practical example illustrates the differences reflected in the final result depending on the chosen method.

5. A Numerical Example

Considering the set of 12 images in Figure 3, each image, except (a), (g), and (j), has at least one target object of the class *cat*, whose ground-truth locations are delimited by the green rectangles. There is a total of 12 target objects limited by the green boxes. Images (b), (e), and (f) have each two ground-truth samples of the target class. An object detector predicted 12 objects represented by the red rectangles (labeled with letters 'A' to 'L') with their associated confidence levels being represented as percentages also shown close to

the corresponding boxes. From the above, images (a), (g), and (j) are expected to have no detection, and images (b), (e), and (f) are expected to have two detections each.

All things considered, to evaluate the precision and recall of the 12 detections it is necessary to establish an IOU threshold t, which will classify each detection as TP or FP. In this example, let us first consider as TP the detections with IOU $> 50\%$, that is $t = 0.5$.

As stated before, AP is a metric that integrates precision and recall in different confidence values. Thus, it is necessary to count the amount of TP and FP classifications given the different confidence levels. Table 3 presents each detection from our example sorted by their confidence levels. In this table, columns $\sum \text{TP}(\tau)$ and $\sum \text{FP}(\tau)$ are the accumulated TPs and FPs, respectively, whose corresponding confidence levels are larger than or equal to the confidence τ specified in the second column of the table. Precision ($Pr(\tau)$) and recall ($Rc(\tau)$) values are calculated based on Equations (4) and (5), respectively. In this example a detection is considered as a TP only if its IOU is larger than 50%, and in this case the column 'IOU > 0.5?' is marked as 'Yes', otherwise it is marked as 'No' and is considered an FP. In this example, all detections overlap some ground-truth with IOU > 0.5, except detection 'J', which is not overlapping any ground-truth, so there is no IOU to be computed in this case.

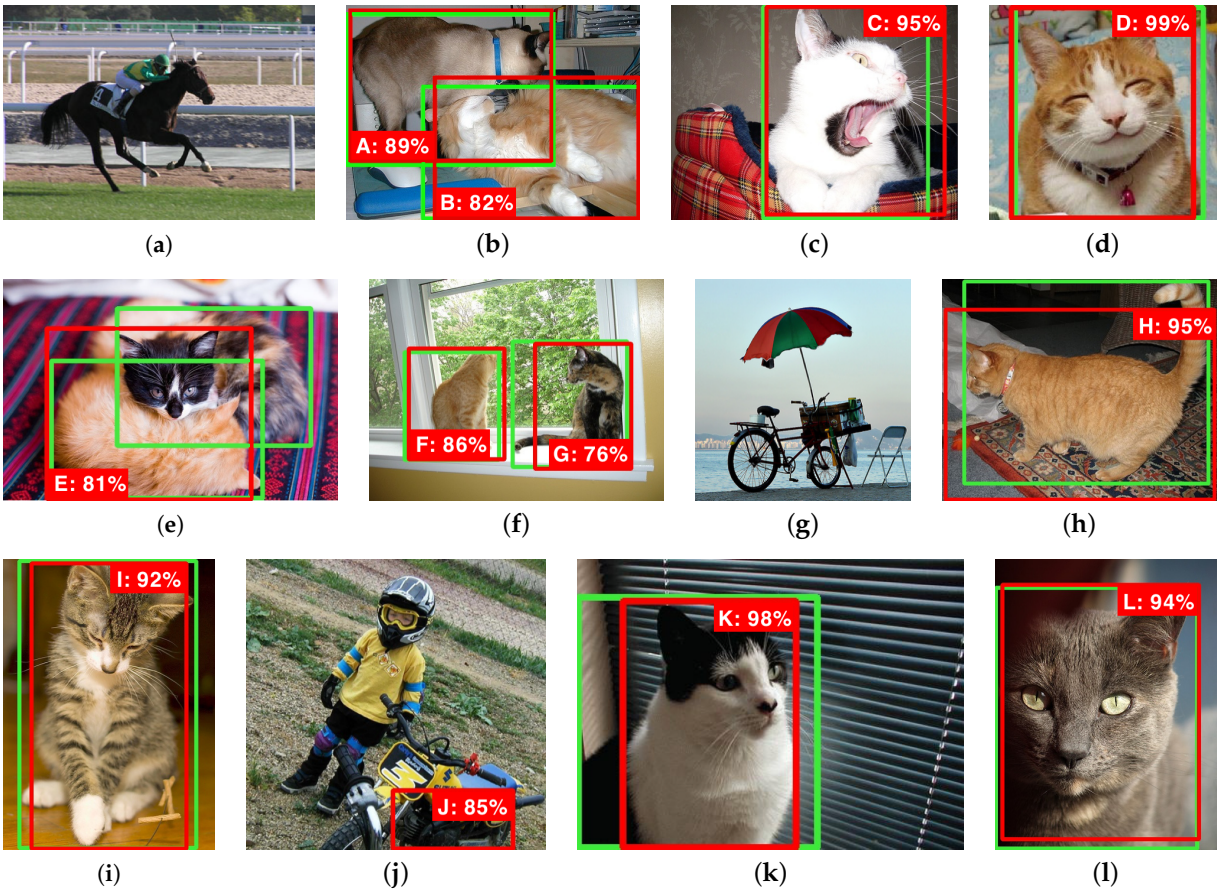


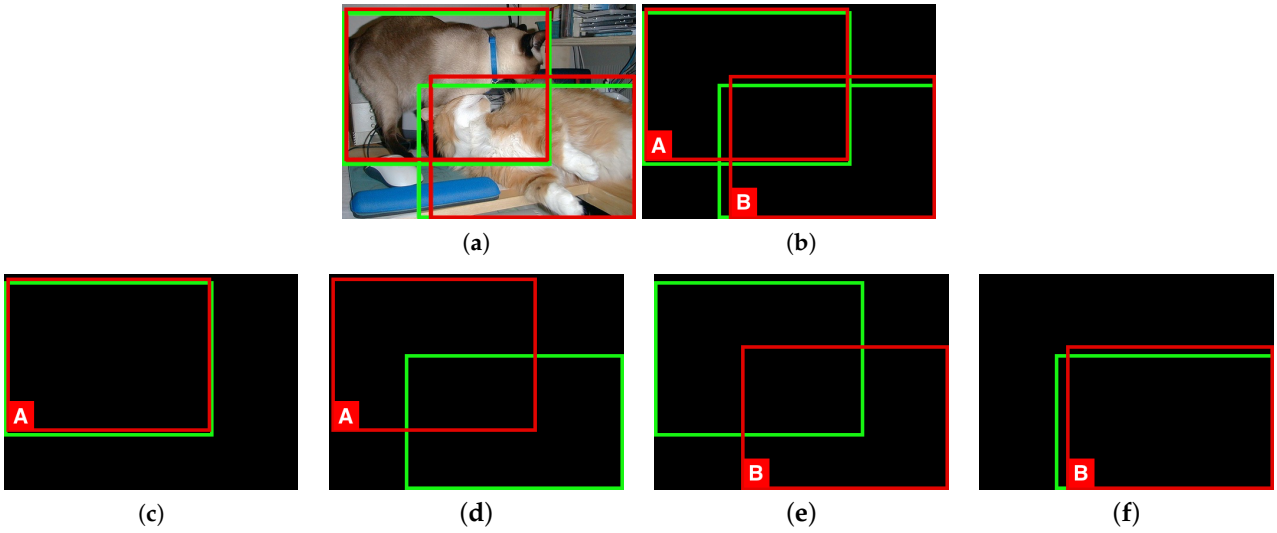
Figure 3. Samples of 12 images from the PASCAL VOC 2012 dataset [67] with ground-truth objects of the class *cat* in green boxes, and the detections performed by [9] in red boxes along with their respected confidence levels. In samples (**b**–**d**,**f**,**h**,**i**,**k**,**l**) the amount of the ground-truth and detected objects is the same. In samples of images (**a**,**g**), no ground-truth object should be detected but one false detection occurred in image (**j**). In sample (**e**) there are two target objects to be detected, but the detector missed one of them.

Some detectors can output one detection overlapping multiple ground truths, as seen in the image from Figure 3b with detections 'A' and 'B'. As detection 'A' has a higher confidence than 'B' (89% $>$ 82%), 'A' has the preference over 'B' to match the ground-truth, so 'A' is associated with the ground truth which gives the highest IOU. Figure 4c,d show

the two possible associations that 'A' can have, ending up with the first one, which presents a higher IOU. Detection 'B' is left with the remaining ground truth in Figure 4f. Another similar situation where one detection could be associated with more than one ground truth is faced by detection 'E' in Figure 3e. The application of the same rule results in matching detection 'E' with the ground truth whose IOU is the highest, represented by the fairer cat, at the bottom of the image.

Table 3. Precision and recall values for detections in Figure 3, that contain a total of 12 ground truths, considering an IOU threshold $t = 0.5$.

Bounding Box	Confidence(τ)	IOU	IOU > 0.5?	$\sum$TP(τ)	$\sum$FP(τ)	$Pr(\tau)$	$Rc(\tau)$
D	99%	0.91	Yes	1	0	1.0000	0.0833
K	98%	0.70	Yes	2	0	1.0000	0.1667
C	95%	0.86	Yes	3	0	1.0000	0.2500
H	95%	0.72	Yes	4	0	1.0000	0.3333
L	94%	0.91	Yes	5	0	1.0000	0.4167
I	92%	0.86	Yes	6	0	1.0000	0.5000
A	89%	0.92	Yes	7	0	1.0000	0.5833
F	86%	0.87	Yes	8	0	1.0000	0.6667
J	85%	-	No	8	1	0.8889	0.6667
B	82%	0.84	Yes	9	1	0.9000	0.7500
E	81%	0.74	Yes	10	1	0.9091	0.8333
G	76%	0.76	Yes	11	1	0.9167	0.9167



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4. Particular cases showing detected bounding boxes overlapping multiple ground truths. (**a**) Original image with predicted (red) and ground-truth (green) bounding boxes. (**b**) Bounding boxes only. (**c**,**d**) Possible overlaps of the first ground truth. (**c**) Detection 'A' overlapping the first ground truth with IOU=.92. (**d**) Detection 'A' overlapping the second ground truth with IOU=.20. (**e**,**f**) Possible overlaps of the second ground truth. (**e**) Detection 'B' overlapping the first ground truth with IOU=.19. (**f**) Detection 'B' overlapping the second ground truth with IOU=.84.

By choosing a more restrictive IOU threshold, different precision $Pr(\tau)$ and recall $Rc(\tau)$ values can be obtained. Table 4 computes the precision and recall values with a more strict IOU threshold of $t = 0.75$. By that, it is noticeable the occurrence of more FP detections and less TP detections, thus reducing both the precision $Pr(\tau)$ and recall $Rc(\tau)$ values.

Graphical representations of the $Pr(\tau) \times Rc(\tau)$ values presented in Tables 3 and 4 can be seen in Figure 5. By comparing both curves, one may note that for this example:

- With a less restrictive IOU threshold ($t = 0.5$), higher recall values can be obtained with the highest precision. In other words, the detector can retrieve about 66.5% of the total ground truths without any miss detection.
- Using $t = 0.75$, the detector is more sensitive to different confidence values τ. This is explained by the more accentuated monotonic behavior for this IOU threshold.
- Regardless the IOU threshold applied, this detector can never retrieve 100% of the ground truths ($Pr(\tau) = 1$) for any confidence value τ. This is due to the fact that the algorithm failed to output any bounding box for one of the ground truths in Figure 3e.

Table 4. Precision and recall values for detections in Figure 3, that contain a total of 12 ground truths, considering an IOU threshold $t = 0.75$.

Bounding Box	Confidence (τ)	IOU	IOU > 0.75?	$\sum$TP(τ)	$\sum$FP(τ)	$Pr(\tau)$	$Rc(\tau)$
D	99%	0.91	Yes	1	0	1.0000	0.0833
K	98%	0.70	No	1	1	0.5000	0.0833
C	95%	0.86	Yes	2	1	0.6667	0.1667
H	95%	0.72	No	2	2	0.5000	0.1667
L	94%	0.91	Yes	3	2	0.6000	0.2500
I	92%	0.86	Yes	4	2	0.6667	0.3333
A	89%	0.92	Yes	5	2	0.7143	0.4167
F	86%	0.87	Yes	6	2	0.7500	0.5000
J	85%	-	No	6	3	0.6667	0.5000
B	82%	0.84	Yes	7	3	0.7000	0.5833
E	81%	0.74	No	7	4	0.6364	0.5833
G	76%	0.76	Yes	8	4	0.6667	0.6667

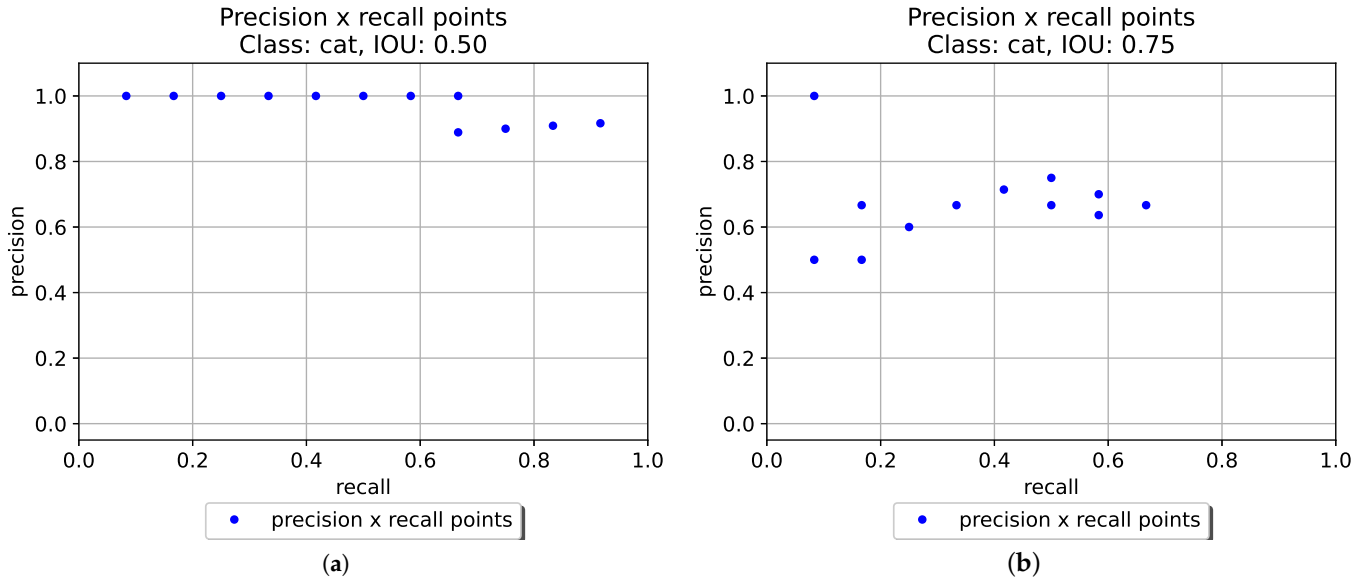


Figure 5. Precision × Recall points with values calculated for: (**a**) Results provided in Table 3. (**b**) Results provided in Table 4.

Note that Figure 5 suggests that an IOU threshold of $t = 0.5$ is less affected by different confidence levels. The graph for the lowest IOU threshold ($t = 0.5$) shows that when confidence levels τ are high, the precision $Pr(\tau)$ does not vary, being equal to the maximum (1.0) for most of confidence values τ. However, in order to detect more objects (increasing the recall $Rc(\tau)$), it is necessary to set a lower confidence threshold τ, which reduces the precision at most by 12%. On the other hand, considering the highest IOU threshold ($t = 0.75$), the detector can retrieve half of the target objects (recall = 0.5) with a precision of 0.75.

As previously explained, different methods can be applied to estimate the average precision, that is, the area under the precision × recall curve. To obtain AP using the N-point interpolation in Equation (11) with $N = 11$ points, the area under the $Pr \times Rc$ curve is computed as the average of the interpolated precision $Pr_{\text{interp}}(R)$ (Equation (9)) samples considering the sampling recall points R at $R_r(n)$ in the set $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ (Equation (10)). On the other hand, to obtain AP using the all-point interpolation approach, the area under the $Pr \times Rc$ curve is computed by the Riemann integral in Equation (9), sampling the recall points R at $R_r(n)$ coincident with the $Rc(\tau)$ values given by the last column of Table 3 or of Table 4 (Equation (12)). The results can be seen in Figure 6. When an IOU threshold $t = 0.5$ was applied, the 11-point interpolation method obtained AP = 88.64% while the all-point interpolation method resulted in a slightly higher AP, reaching AP = 89.58%. Similarly, for an IOU threshold $t = 0.75$, the 11-point interpolation method obtained AP = 49.24% and the all-point interpolation obtained AP = 50.97%.

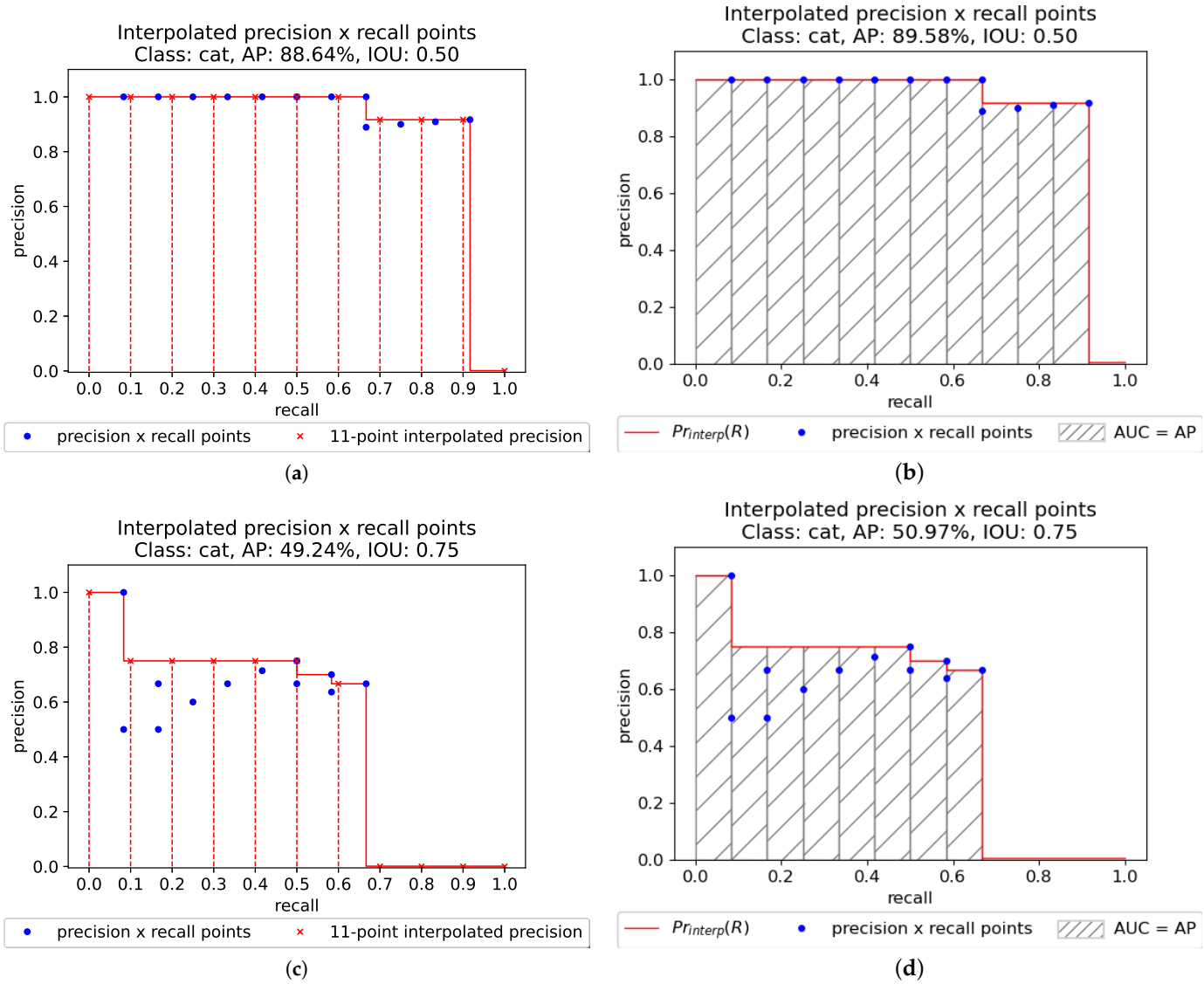


Figure 6. Results for different approaches for computing the AP metric. (**a**) 11-point interpolation with IOU threshold $t = 0.5$. (**b**) All-point interpolation with IOU threshold $t = 0.5$. (**c**) 11-point interpolation with IOU threshold $t = 0.75$. (**d**) All-point interpolation with IOU threshold $t = 0.75$.

When a lower IOU threshold t was considered ($t = 0.5$ as opposed to $t = 0.75$), the AP was considerably increased in both interpolation approaches. This is caused by the increase in the TP detections, due to a lower IOU threshold.

If focus is shifted towards how well localized the detections are, irrespective of their confidence values, it is sensible to consult the AR metrics (Equations (14)–(18)). Computing twice the average excess IOU for the samples in this practical example as in Equation (15), yields AR = 60%, while computing the average max recall across the standard COCO IOU thresholds, that is $t \in \{0.50, 0.55, \ldots, 0.95\}$, as in Equation (17), yields AR = 66%. As the latter computation effectively does a coarser quantization of the IOU space, the two AR figures differ slightly. The next section enlists and briefly describes which variations of the metrics based on AP and AR are more frequently employed in the literature. In most cases they are the result of combinations of different IOU thresholds and interpolation methods.

6. Most Employed Metrics Based on AP and AR

As previously presented, there are different ways to evaluate the area under the precision × recall and recall × IOU curves. Nonetheless, besides such combinations of different IOU thresholds and interpolation points, that are other variations that result in different metric values. Some methods limit the evaluation by object scales and detections per image. This section overviews the distinctions behind all the metrics shown in Table 2.

6.1. AP with IOU Threshold $t = 0.5$

This AP metric is widely used to evaluate detections in the PASCAL VOC dataset [67]. Its official implementation is in MATLAB and it is available in the PASCAL VOC toolkit. It measures the AP of each class individually by computing the area under the precision × recall curve interpolating all points as presented in Equation (9). In order to classify detections as TP or FP the IOU threshold is set to $t = 0.5$.

6.2. mAP with IOU Threshold $t = 0.5$

This metric is also used by the PASCAL VOC dataset and is also available in their MATLAB toolkit. It is calculated as the AP with IOU $t = 0.5$, but the result obtained by each class is averaged as given in Equation (13).

6.3. AP@.5 and AP@.75

These two metrics evaluate the precision × recall curve differently than the PASCAL VOC metrics. In this method, the interpolation is performed in $N = 101$ recall points, as given in Equation (11). Then, the computed results for each class are summed up and divided by the number of classes, as in Equation (13).

The only difference between AP@.5 and AP@.75 regards the applied IOU thresholds. AP@.5 uses $t = 0.5$ whereas AP@.75 applies $t = 0.75$. These metrics are commonly used to report detections performed in the COCO dataset and are officially available in their official evaluation tool.

6.4. AP@[.5:.05:.95]

This metric expands the AP@.5 and AP@.75 metrics by computing the AP@ with 10 different IOU thresholds ($t = [0.5, 0.55, ..., 0.95]$) and taking the average among all computed results.

6.5. AP_S, AP_M, and AP_L

These three metrics, also referred to as AP Across Scales, apply the AP@[.5,.05:.95] from Section 6.1 taking into consideration the area of the ground-truth object:

- AP_S only evaluates small ground-truth objects (area < 32^2 pixels);
- AP_M only evaluates medium-sized ground-truth objects (32^2 < area < 96^2 pixels);
- AP_L only evaluates large ground-truth objects (area > 96^2).

When evaluating objects of a given size, objects of the other sizes (both ground-truth and predicted) are not considered in the evaluation. This metric is also part of the COCO evaluation dataset.

6.6. AR_1, AR_{10}, and AR_{100}

These AR variations apply Equation (14) limiting the number of detections per image, that is, they calculate the AR given a fixed amount of detections per image, averaged over all classes and IOUs. The IOUs used to measure the recall values are the same as in AP@[.5,.05:.95].

AR_1 considers up to one detection per image, while AR_{10} and AR_{100} consider at most 10 and 100 objects per image, respectively.

6.7. AR_S, AR_M and AR_L

Similarly to the AR variations with limited number of detections per image, these metrics evaluate detections considering the same areas as the AP across scales. As the metrics based on AR are implemented in the COCO official evaluation tool, they are regularly reported with the COCO dataset.

6.8. F1-Score

The F1-score is defined as the harmonic mean of the precision (Pr) and recall (Rc) of a given detector, that is:

$$\mathrm{F}_1 = 2\frac{Pr.Rc}{Pr + Rc} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{\mathrm{FN}+\mathrm{FP}}{2}}. \tag{19}$$

The F1-score is limited to the interval [0, 1], being 0 if precision or recall (or both) are 0, and 1 when both precision and recall are 1.

As the F1-score does not take into account different confidence values, it is only used to compare object detectors in a fixed confidence threshold level τ.

6.9. Other Metrics

Other less popular metrics have also been proposed to evaluate object detections. They are mainly designed to be applied with particular datasets. The Open Images Object Detection Metric, for example, is similar to mAP (IOU=.50), being specifically designed to consider special ground-truth annotations of the Open Images dataset [68]. This dataset groups into a single annotation five or more objects of the same class that somehow are occluding each other, such as *a group of flowers* or *a group of people*. This metric simply ignores a detection if it overlaps a ground-truth box tagged as *group of*, whose area of intersection between the detection and ground-truth boxes divided by the area of the detection is greater than 0.5. This way, it does not penalize detections matching a group of very close ground-truth objects.

The localization recall-precision (LRP) error, a new metric suggested in [91], intends to consider the accuracy of the detected bounding box localization and equitably evaluate situations where the AP is unable to distinguish very different precision $\times$ recall curves.

6.10. Comparisons among Metrics

In practice, the COCO's AP@[.5:.05:.95] and PASCAL mAP metrics are the most popular ones used as benchmarks. However, as COCO's AP@[.5:.05:.95] is affected by different IOUs, it is not possible to evaluate the effectiveness of the detector with a more or less restrictive IOU with this metric. For a more strict evaluation with respect to the likeness of the ground truth and detection bounding boxes, the AP@.75 metric should be applied. In datasets where the objects appear to have relatively different sizes, AP metrics concerning their areas should be employed. By that, the assertiveness of objects with similar relative sizes can be compared. As shown in this work, the interpolation methods applied by the AP metrics try to remove the non-monotonic behavior of the $Pr(\tau) \times Rc(\tau)$ curve before calculating its AUC. In an N-point interpolation, a greater N leads to a better AUC approximation. Therefore, the 101-point interpolation approach used by COCO's AP metrics provides a better AUC approximation than the 11-point interpolation approach. On the other hand, PASCAL VOC uses the all-point interpolation, which is an even better approximation of the AUC. In cases where the detector is expected to detect at least a

certain amount of objects in a given image (e.g., detecting one bird in a flock of birds should be sufficient), AR metrics regarding detections or sizes are more appropriate.

7. Evaluating Object Detection in Videos

Many works in the literature use the mAP as a metric to evaluate the performance of object detection models in video sequences [92–98]. In this case, the frames are evaluated independently, ignoring the spatio-temporal characteristics of the objects presented in the scene. The authors of [92] categorized the ground-truth objects according to their motion speed. This is done by measuring the average IOU score of the current frame and the nearby ± 10 frames. In this context, in addition to mAP, they also reported the mAP over the slow, medium, and fast groups, denoted as mAP(slow), mAP(medium), and mAP(fast), respectively.

In some applications, the latency in the identification of the objects of interest plays a crucial role in how well the overall system will perform. Detection delay, defined as the number of frames between the first occurrence of an object in the scene and its first detection, then becomes an important measurement for time-critical systems. The authors in [99] claim that AP is not sufficient to quantify the temporal behavior of detectors, and propose a complementary metric, the average delay (AD), averaging the mean detection delay over multiple false positive ratio thresholds, and over different object sizes, yielding a metric that fits well for systems that rely on timely detections. While the cost of detection latency is significant for a somewhat niche set of tasks, the inclusion of time information for video detection metrics can be useful to assess system behaviors that would otherwise be elusive when only the standard, frame level AP metric is used.

7.1. Spatio-Temporal Tube Average Precision

As discussed above, the aforementioned metrics are all used on an image or frame level. However, when dealing with videos, one may be interested in evaluating the model performance at the whole video level. In this work, we propose an extension of the AP metric to evaluate video object detection models that we refer to as spatio-temporal tube AP (STT-AP). As in AP, a threshold over the IOU is also used to determine whether the detections are correct or not. However, instead of using two types of overlaps (spatial and temporal), we extend the standard IOU definition to consider the spatio-temporal tubes generated by the detection and of the ground truth. This metric, that integrates spatial and temporal localizations, is concise, yet expressive.

Instead of considering each detection of the same object independently along the frames, the spatial bounding boxes of the same object are concatenated along the temporal dimension, forming a spatio-temporal tube, which is the video analogous to an image bounding box. A spatio-temporal tube T_o of an object o is the spatio-temporal region defined as the concatenation of the bounding boxes of this object from each frame of a video, that is:

$$T_o = \left[B_{o,q} B_{o,q+1} \cdots B_{o,q+Q-1}\right], \tag{20}$$

where $B_{o,k}$ is the bounding box of the object o in frame k of the video that is constituted of Q frames indexed by $k = q, q+1, \ldots, q+Q-1$.

Using spatio-temporal tubes, the concept of IOU used in object detection in images (see Section 4) can be naturally extended to videos. Considering a ground-truth spatio-temporal tube T_{gt} and a predicted spatio-temporal tube T_p, the spatio-temporal tube IOU (STT-IOU) measures the ratio of the overlapping to the union of the "discrete volume" between T_{gt} and T_p, such that:

$$\text{STT-IOU} = \frac{\text{volume}(T_p \cap T_{gt})}{\text{volume}(T_p \cup T_{gt})} = \frac{\sum_k \text{area of overlap in frame } k}{\sum_k \text{area of union in frame } k}, \tag{21}$$

as illustrated in Figure 7.

$$\text{STT-IOU} = \frac{\text{volume of overlap}}{\text{volume of union}} =$$

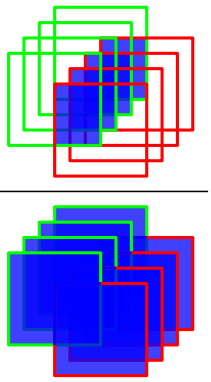


Figure 7. Spatio-temporal tube IOU (STT-IOU).

In this way, an object is considered a TP if the STT-IOU is equal or greater than a chosen threshold. As in the conventional AP, this metric may be as rigorous as desired. The closer to 1 it is, the more well-located the predicted tube must be to be considered a TP.

Figure 8 illustrates four different STTs. The STTs in red are detected STTs and the STTs in green are ground-truth STTs. The STT in (1) constitutes an FP case, and the STT in (2) may be a TP case (depending on the STT-IOU) , as it intersects the corresponding ground-truths STT (3). Since there is no detection corresponding to the ground-truth STT (4), it is an FN.

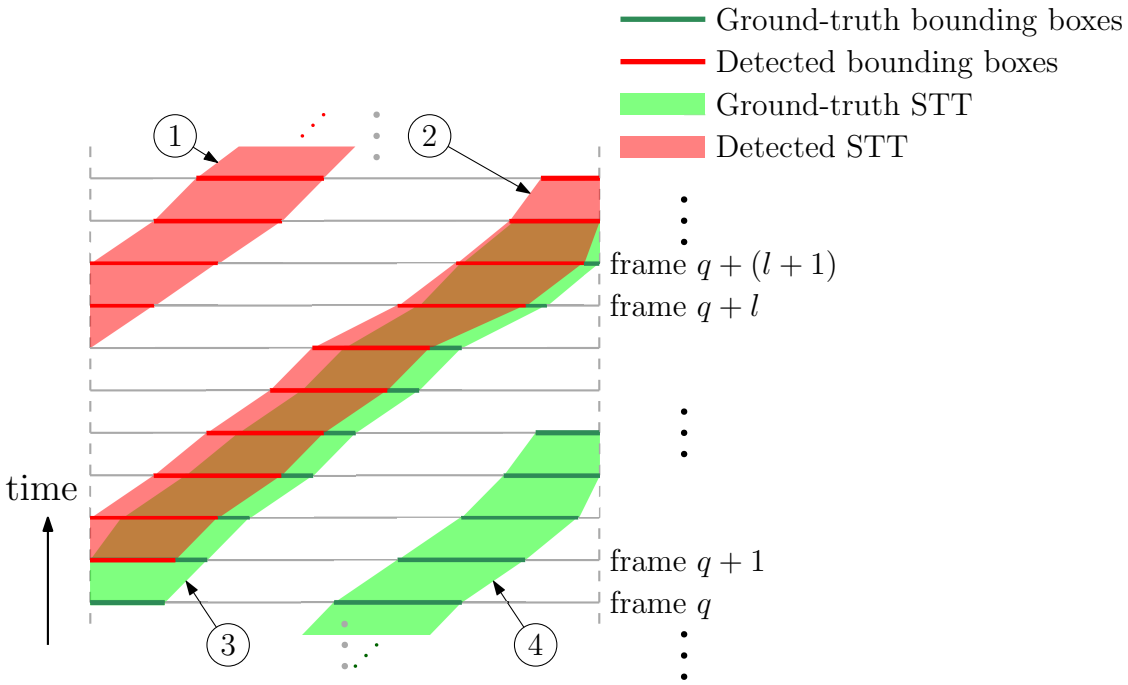


Figure 8. Illustration of STTs. The STT (1) constitutes a false positive (FP). The STT (2) may constitute true positive (TP) (depending on the STT-IOU) as the detection STT intersects the corresponding ground-truth STT (3). The STT (4) constitutes a false negative (FN).

Based on these definitions, the proposed STT-AP metric follows the AP: For each class, each ground-truth tube is associated with the predicted tube of the same class that presents the highest STT-IOU scores (since it is higher than the threshold). The ground-truth tubes not associated with a predicted tube are FNs and the predicted tubes not associated with a ground-truth tube are FPs. Then, the spatio-temporal tube predictions are ranked according to the predicted confidence level (from the highest to the lowest), irrespective of correctness. Since the STT-AP evaluates the detection of an object in the video as a whole, the confidence level assumed for a spatio-temporal tube is the average confidence of the bounding boxes corresponding to each of its constituent frames. After that, the all-point interpolation

(Section 4.2.2) is performed allowing one to compute the proposed STT-AP. This procedure may be repeated and averaged for all classes in the database, yielding the so-called mean STT-AP (mSTT-AP). From Equation (21) one can readily see that the computational cost per frame of the SST-AP is similar to the one of the frame-by-frame mAP.

8. An Open-Source Toolbox

This paper focuses on explaining and comparing the different metrics and formats currently used in object detection, detailing the specifications and pointing out the particularities of each metric variation. The existing tools provided by popular competitions [24,81–85] are not adequate to evaluate metrics using annotations in formats that are different from their native ones. Thus, to complement the analysis of the metrics presented here, the authors have developed and released an open-source toolkit as a reliable source of object detection metrics for the academic community and researchers.

With more than 3100 stars and 740 forks, our previously available tool for object detection assessment [100] has received positive feedback from the community and researchers. It has also been used as the official tool in competition [86], adopted in 3rd-party libraries such as [101], and parts of our code have been used by many other works such as in YoloV5 [9]. Besides the significant acceptance by the community, we have received many requests to expand the tool in order to support new metrics and bounding box formats. Such demands motivated us to offer more evaluation metrics, to accept more bounding box formats, and to present a novel metric for object detection in videos.

This tool implements the same metrics used by the most popular competitions and object-detection benchmark researches. This implementation does not require modifications of the detection model to match complicated input formats, avoiding conversions to XML, JSON, CSV, or other file types. It supports more than eight different kinds of annotation formats, including the ones presented in Table 1. To ensure the accuracy of the results, the implementation strictly followed the metric definitions and the output results were carefully validated against the ones of the official implementations.

Developed in Phython and supporting 14 object detection metrics for images, this work also incorporates the novel spatio-temporal metric described in Section 7.1 to evaluate detected object in videos aggregating some of the concepts applied to evaluate detections in images. From a practical point of view, the tool can also be adapted and expanded to support new metrics and formats. The expanded project distributed with this paper can be accessed at https://github.com/rafaelpadilla/review_object_detection_metrics [102].

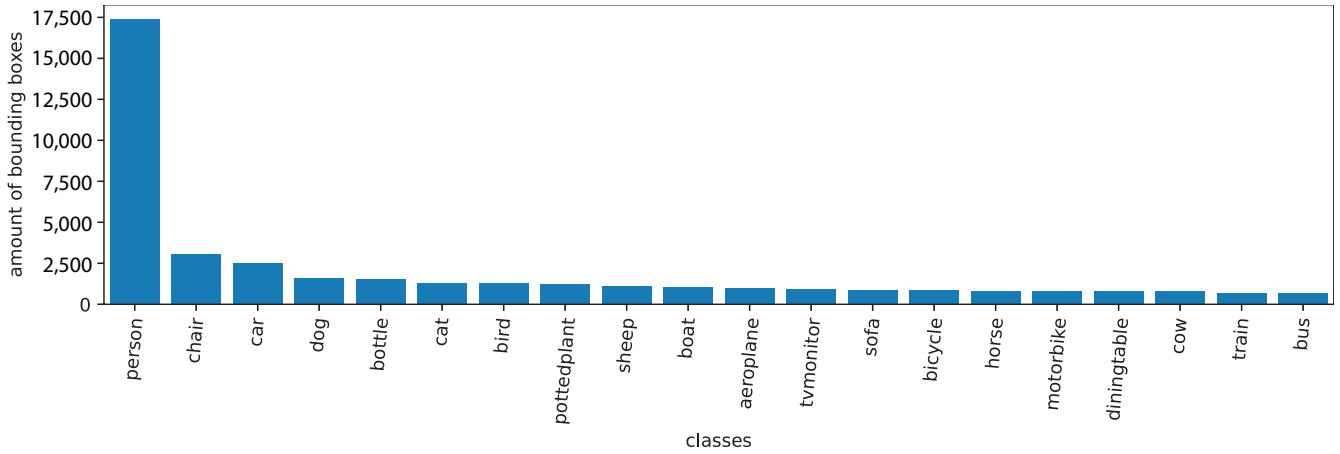
9. Metrics Evaluation in a Practical Example

In this section, we use different object detection metrics to evaluate YOLOv5 model [9]. The chosen model was trained with the COCO dataset and was applied in the training/validation PASCAL VOC 2012 dataset [67]. Intentionally different datasets were used to train and evaluate the model to evidence the potential of our tool to deal with different ground-truth and detected bounding-box formats. For this experiment, the annotations of the ground-truth boxes are in PASCAL VOC format containing 20 classes of objects, while the model was trained with COCO dataset and was able to detect objects in 80 classes, predicting detections in text files in the YOLO format.

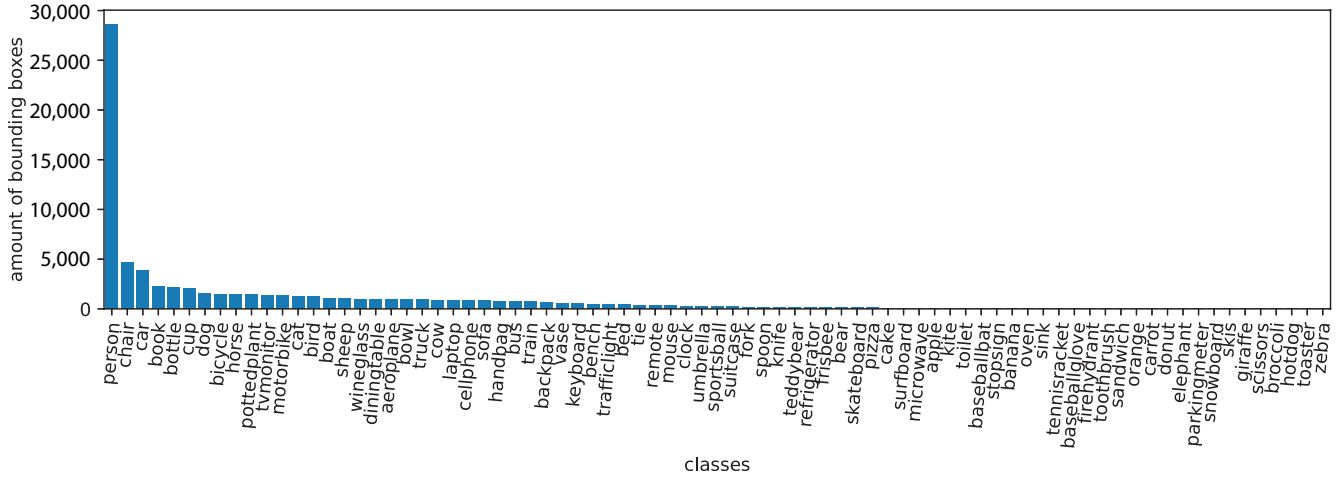
By using our tool, one can quickly obtain 14 different metrics without the necessity to convert files to specific formats. As some classes of the ground-truth dataset are tagged differently by the detector (e.g., PASCAL VOC class *tvmonitor* is referred to as *tv* in COCO dataset), the only required work is to provide a text file listing the names of the classes in the ground-truth format. This way the evaluation tool can recognize that the detected object *airplane* should be evaluated as *aeroplane*.

A total of 17,125 of images from the train/val PASCAL VOC 2012 dataset containing 40,138 objects of 20 classes were evaluated by the YOLOV5 model to detect objects in 80 different classes. A total of 74,752 detections were detected by the model. Figure 9 compares the distribution of ground-truth and detected objects per class. Due to the difference of

classes in the training and testing datasets, many predicted classes are not in the ground-truth set, so detections of the extra classes are ignored by the metrics.



(**a**) Class distribution in the ground-truth dataset.



(**b**) Class distribution of the detected objects.

Figure 9. Class distributions: (**a**) Ground-truth bounding boxes. (**b**) Detected bounding boxes.

The AP results for each class are presented in Table 5. The highest AP values over all classes were obtained when the AUC was measured with the 11-point interpolation method and an IOU threshold of $t = 0.5$, resulting in $mAP = 0.58$. As expected for all cases, a more rigorous IOU threshold ($t = 0.75$) resulted in a smaller AP. Comparing the individual AP results among all classes, the most difficult object for all interpolation methods was the *potted plant*, having an AP not higher than 0.37 for an IOU threshold of $t = 0.5$ and an AP not higher than 0.22 with an IOU threshold of $t = 0.75$.

Table 5. AP results obtained with different interpolation methods and IOU thresholds.

Class	IOU Threshold = 0.5			IOU Threshold = 0.75		
	101-Point	**11-Point**	**All-Point**	**101-Point**	**11-Point**	**All-Point**
aeroplane	0.76	**0.79**	0.77	0.57	0.58	0.58
bicycle	0.41	0.43	0.41	0.31	0.33	0.31
bird	0.66	0.67	0.66	0.49	0.48	0.50
boat	0.47	0.46	0.47	0.28	0.29	0.29
bottle	0.45	0.47	0.45	0.32	0.34	0.33
bus	0.79	0.78	0.80	0.74	0.69	0.74
car	0.52	0.53	0.53	0.39	0.39	0.39
cat	0.73	0.74	0.73	0.54	0.53	0.54
chair	0.41	0.40	0.41	0.30	0.32	0.30
cow	0.74	0.69	0.74	0.59	0.58	0.60
diningtable	0.44	0.46	0.44	0.28	0.31	0.28
dog	0.66	0.64	0.65	0.53	0.53	0.53
horse	0.42	0.43	0.43	0.35	0.36	0.35
motorbike	0.51	0.53	0.51	0.38	0.39	0.38
person	0.67	0.65	0.68	0.53	0.54	0.53
pottedplant	0.37	0.39	0.37	0.22	0.25	0.22
sheep	0.68	0.68	0.68	0.56	0.58	0.57
sofa	0.44	0.45	0.44	0.37	0.38	0.37
train	0.75	0.77	0.76	0.65	0.66	0.65
tvmonitor	0.54	0.55	0.54	0.43	0.45	0.43
average	0.57	0.58	0.57	0.44	0.45	0.44

The results obtained by the variations which apply AP and AR with different sizes and quantity of objects per image are summarized in Table 6.

Table 6. Values of AP and average recall (AR) variations for different object sizes and number of detections per image.

Metric	Result
AP_S	0.13
AP_M	0.33
AP_L	0.46
AR_1	0.39
AR_{10}	0.53
AR_{100}	0.53
AR_S	0.23
AR_M	0.47
AR_L	0.58

Even if the same interpolation technique is applied, the results may vary depending on the IOU threshold. Similarly, different interpolations with the same IOU threshold may also lead to distinct results.

The metrics considering objects in different scales are useful to compare the assertiveness of detections in datasets containing objects of different scales. In the COCO dataset, for instance, roughly 42% of the objects are considered small (area < 32^2 pixels), 34% are considered medium (32^2 < area < 96^2 pixels), and 24% are considered large (area > 96^2 pixels). This explains the vast amount of works using this dataset to report their results.

10. Conclusions

This work analyzed the formats of bounding boxes used to represent the objects in popular datasets, demonstrated the most common benchmark object detection metrics, and suggested a new metric for videos, the spatio-temporal tube average precision (STT-AP), based on the concepts used to evaluate object detection in images. The similarities and

inconsistencies of each metric were examined, and our results revealed their dissimilarities by evaluating the predictions of a pre-trained object detector in a largely used dataset. A toolkit implementing all described metrics in a way compatible to most data-annotation formats in use was presented and validated. Such results may facilitate direct and unified comparisons among most algorithms being proposed in the field of object detection. For future work, we intend to perform a regular survey update through its companion website, incorporating newly proposed metrics and annotation formats, and extend it to the problem of object tracking.

Author Contributions: Conceptualization, all authors; methodology, all authors; software, R.P., W.L.P. and T.L.B.D.; validation, R.P., W.L.P. and T.L.B.D.; formal analysis, all authors; investigation, all authors; writing—original draft preparation, R.P., W.L.P. and T.L.B.D.; writing—review and editing, S.L.N. and E.A.B.d.S.; visualization, R.P., W.L.P. and T.L.B.D.; supervision, S.L.N. and E.A.B.d.S.; funding acquisition, S.L.N. and E.A.B.d.S. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ). Wesley L. Passos and Eduardo A. B. da Silva are also partially funded by the Google Latin America Research Awards (LARA) 2020.

Data Availability Statement: The images used in Section 5 are part of the PASCAL VOC dataset and can be downloaded at http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html. The model presented in Section 9 was trained with the COCO dataset, which can be downloaded at https://cocodataset.org, and tested with images from the PASCAL VOC dataset, which can be downloaded at http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html. The tool presented in this work can be accessed at https://github.com/rafaelpadilla/review_object_detection_metrics.

Acknowledgments: The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cheng, F.C.; Huang, S.C.; Ruan, S.J. Illumination-Sensitive Background Modeling Approach for Accurate Moving Object Detection. *IEEE Trans. Broadcast.* **2011**, *57*, 794–801. [CrossRef]
2. Khan, F.S.; Anwer, R.M.; Van De Weijer, J.; Bagdanov, A.D.; Vanrell, M.; Lopez, A.M. Color Attributes for Object Detection. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3306–3313.
3. Ouyang, W.; Wang, X. A Discriminative Deep Model for Pedestrian Detection with Occlusion Handling. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3258–3265.
4. Gao, T.; Packer, B.; Koller, D. A Segmentation-Aware Object Detection Model with Occlusion Handling. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 1361–1368.
5. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *arXiv* **2019**, arXiv:1905.05055.
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
7. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
9. Jocher, G.; Stoken, A.; Borovec, J.; NanoCode012, C.; Changyu, L.; Laughing, H. ultralytics/yolov5: v3.0. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 December 2020).
10. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 743–761. [CrossRef]
11. Ohn-Bar, E.; Trivedi, M.M. To Boost or Not to Boost? On the Limits of Boosted Trees for Object Detection. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 3350–3355.

12. Viola, P.; Jones, M. Robust Real-Time Object Detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
13. Hirano, Y.; Garcia, C.; Sukthankar, R.; Hoogs, A. Industry and Object Recognition: Applications, Applied Research and Challenges. *Towar. Categ. Level Object Recognit.* **2006**, *4170*, 49–64.
14. Wang, X. Intelligent Multi-Camera Video Surveillance: A Review. *Pattern Recognit. Lett.* **2013**, *34*, 3–19. [CrossRef]
15. Franke, K.; Srihari, S.N. Computational Forensics: An Overview. In Proceedings of the International Workshop on Computational Forensics (IWCF), Washington, DC, USA, 7–8 August 2008; pp. 1–10.
16. Baltieri, D.; Vezzani, R.; Cucchiara, R. 3DPes: 3D People Dataset for Surveillance and Forensics. In Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding, Scottsdale, AZ, USA, 28 November–1 December 2011; pp. 59–64.
17. Olabarriaga, S.D.; Smeulders, A.W. Interaction in the Segmentation of Medical Images: A Survey. *Med Image Anal.* **2001**, *5*, 127–142. [CrossRef]
18. Cootes, T.F.; Taylor, C.J. Statistical Models of Appearance for Medical Image Analysis and Computer Vision. In Proceedings of the Medical Imaging 2001: Image Processing, San Diego, CA, USA, 3 July 2001; pp. 236–248.
19. Ganster, H.; Pinz, P.; Rohrer, R.; Wildling, E.; Binder, M.; Kittler, H. Automated Melanoma Recognition. *IEEE Trans. Med Imaging* **2001**, *20*, 233–239. [CrossRef]
20. Janai, J.; Güney, F.; Behl, A.; Geiger, A. *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art*; Now Publishers: Norwell, MA, USA; Delft, The Netherlands, 2020.
21. Buch, N.; Velastin, S.A.; Orwell, J. A Review of Computer Vision Techniques for the Analysis of Urban Traffic. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 920–939. [CrossRef]
22. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef]
23. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 27th International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
24. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
25. Attneave, F.; Arnoult, M.D. The Quantitative Study of Shape and Pattern Perception. *Psychol. Bull.* **1956**, *53*, 452–471. [CrossRef] [PubMed]
26. Roberts, L.G. Machine Perception of Three-Dimensional Solids. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1963.
27. Lowe, D.G.; Binford, T.O. The Recovery of Three-Dimensional Structure from Image Curves. *IEEE Trans. Pattern Anal. Mach. Intell.* **1985**, *7*, 320–326. [CrossRef] [PubMed]
28. Harris, C.G.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 23.1–23.6.
29. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, BC, Canada, 24–28 August 1981; pp. 674–679.
30. Shi, J.; Tomasi. Good Features to Track. In Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
31. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
32. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In Proceedings of the 9th European Conference on Computer Vision (ECCV), Graz, Austria, 7–13 May 2006; pp. 404–417.
33. Nguyen, T.; Park, E.A.; Han, J.; Park, D.C.; Min, S.Y. Object Detection Using Scale Invariant Feature Transform. In *In Genetic and Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 65–72.
34. Zhou, H.; Yuan, Y.; Shi, C. Object Tracking Using SIFT Features and Mean Shift. *Comput. Vis. Image Underst.* **2009**, *113*, 345–352. [CrossRef]
35. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
36. Mizuno, K.; Terachi, Y.; Takagi, K.; Izumi, S.; Kawaguchi, H.; Yoshimoto, M. Architectural Study of HOG Feature Extraction Processor for Real-Time Object Detection. In Proceedings of the IEEE Workshop on Signal Processing Systems, Quebec City, QC, Canada, 17–19 October 2012; pp. 197–202.
37. Sun, Z.; Bebis, G.; Miller, R. On-Road Vehicle Detection: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 694–711.
38. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
39. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3350–3355.
41. Hinton, G.E.; Osindero, S.; Teh, Y.W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]

42. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef]
43. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.Y.; Shlens, J.; Le, Q.V. Learning Data Augmentation Strategies for Object Detection. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 566–583.
44. Loey, M.; Manogaran, G.; Khalifa, N.E.M. A Deep Transfer Learning Model with Classical Data Augmentation and CGAN to Detect COVID-19 from Chest CT Radiography Digital Images. *Neural Comput. Appl.* **2020**, 1–13. [CrossRef] [PubMed]
45. González, R.E.; Munoz, R.P.; Hernández, C.A. Galaxy Detection and Identification Using Deep Learning and Data Augmentation. *Astron. Comput.* **2018**, *25*, 103–109. [CrossRef]
46. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014; pp. 1–16.
47. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
48. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
49. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
50. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
51. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020** arXiv:2004.10934.
52. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; p. 587.
53. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
54. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection Via Region-Based Fully Convolutional Networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS), Barcelona, Spain, 5–10 December 2016; pp. 379–387.
55. Gu, J.; Hu, H.; Wang, L.; Wei, Y.; Dai, J. Learning Region Features for Object Detection. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 392–406.
56. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation Networks for Object Detection. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3588–3597.
57. Połap, D. An Adaptive Genetic Algorithm as a Supporting Mechanism for Microscopy Image Analysis in a Cascade of Convolution Neural Networks. *Appl. Soft Comput. J.* **2020**, *97*, 1–11. [CrossRef]
58. S, I.T.J.; Sasikala, J.; Juliet, D.S. Optimized Vessel Detection in Marine Environment Using Hybrid Adaptive Cuckoo Search Algorithm. *Comput. Electr. Eng.* **2019**, *78*, 482–492. [CrossRef]
59. Li, Y.; Wang, H.; Dang, L.M.; Nguyen, T.N.; Han, D.; Lee, A.; Jang, I.; Moon, H. A Deep Learning-based Hybrid Framework for Object Detection and Recognition in Autonomous Driving. *IEEE Access* **2020**, *8*, 194228–194239. [CrossRef]
60. Chen, Y.; Zhou, W. Hybrid-Attention Network for RGB-D Salient Object Detection. *Appl. Sci.* **2020**, *10*, 5806. [CrossRef]
61. Zhang, P.; Liu, W.; Lei, Y.; Lu, H. Hyperfusion-Net: Hyper-Densely Reflective Feature Fusion for Salient Object Detection. *Pattern Recognit.* **2019**, *93*, 521–533. [CrossRef]
62. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; Van Der Laak, J.A.; Van Ginneken, B.; Sánchez, C.I. A Survey on Deep Learning in Medical Image Analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef]
63. Cao, Z.; Duan, L.; Yang, G.; Yue, T.; Chen, Q.; Fu, H.; Xu, Y. Breast Tumor Detection in Ultrasound Images Using Deep Learning. In *International Workshop on Patch-Based Techniques in Medical Imaging*; Springer: Cham, Switzerland, 2017; pp. 121–128.
64. Jaeger, P.F.; Kohl, S.A.; Bickelhaupt, S.; Isensee, F.; Kuder, T.A.; Schlemmer, H.P.; Maier-Hein, K.H. Retina U-Net: Embarrassingly Simple Exploitation of Segmentation Supervision for Medical Object Detection. In Proceedings of Machine Learning for Health Workshop (ML4H), Vancouver, BC, Canada, 13–14 December 2020; pp. 171–183.
65. Li, Z.; Dong, M.; Wen, S.; Hu, X.; Zhou, P.; Zeng, Z. CLU-CNNs: Object detection for Medical Images. *Neurocomputing* **2019**, *350*, 53–59. [CrossRef]
66. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
67. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. Available online: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (accessed on 20 December 2020).
68. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]

69. Wada, K. labelme: Image Polygonal Annotation with Python. 2016. Available online: https://github.com/wkentaro/labelme (accessed on 20 December 2020).
70. Lin, T. LabelImg. 2015. Available online: https://github.com/tzutalin/labelImg (accessed on 20 December 2020).
71. Wada, K. VoTT: Visual Object Tagging Tool. Available online: https://github.com/Microsoft/VoTT (accessed on 20 December 2020).
72. Sekachev, B.; Manovich, N.; Zhiltsov, M.; Zhavoronkov, A.; Kalinin, D. opencv/cvat v1.1.0. 2020. Available online: http://doi.org/10.5281/zenodo.4009388 (accessed on 20 December 2020).
73. Dutta, A.; Zisserman, A. The VIA Annotation Software for Images, Audio and Video. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 2276–2279.
74. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
75. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: https://www.tensorflow.org/ (accessed on 20 December 2020).
76. Law, H.; Deng, J. Cornernet: Detecting Objects as Paired Keypoints. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
77. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
78. Liu, S.; Huang, D. Receptive Field Block Net for Accurate and Fast Object Detection. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
79. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-Shot Refinement Neural Network for Object Detection. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4203–4212.
80. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
81. Open Images Object Detection RVC 2020 Edition. Available online: https://www.kaggle.com/c/open-images-object-detection-rvc-2020 (accessed on 20 December 2020).
82. COCO Detection Challenge (Bounding Box). Available online: https://competitions.codalab.org/competitions/20794 (accessed on 20 December 2020).
83. Datalab Cup: CNN Object Detection. Available online: https://www.kaggle.com/c/datalabcup-cnn-object-detection (accessed on 20 December 2020).
84. Google AI Open Images-Object Detection Track. Available online: https://www.kaggle.com/c/google-ai-open-images-object-detection-track (accessed on 20 December 2020).
85. Lyft 3D Object Detection for Autonomous Vehicles. Available online: https://www.kaggle.com/c/3d-object-detection-for-autonomous-vehicles/ (accessed on 20 December 2020). .
86. City Intelligence Hackathon. Available online: https://belvisionhack.ru (accessed on 20 December 2020).
87. Dudczyk, J.; Kawalec, A. Identification of Emitter Sources in the Aspect of Their Fractal Features. *Bull. Pol. Acad. Sci. Tech. Sci.* **2013**, *61*, 623–628. [CrossRef]
88. Rybak, Ł.; Dudczyk, J. A Geometrical Divide of Data Particle in Gravitational Classification of Moons and Circles Data Sets. *Entropy* **2020**, *22*, 1088–1103. [CrossRef] [PubMed]
89. Jaccard, P. Étude Comparative de la Distribution Florale Dans Une Portion des Alpes et des Jura. *Bull. Soc. Vaudoise Des Sci. Nat.* **1901**, *37*, 547–579.
90. Hosang, J.; Benenson, R.; Dollár, P.; Schiele, B. What Makes for Effective Detection Proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 814–830. [CrossRef]
91. Oksuz, K.; Can Cam, B.; Akbas, E.; Kalkan, S. Localization Recall Precision (LRP): A New Performance Metric for Object Detection. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 504–519.
92. Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; Wei, Y. Flow-Guided Feature Aggregation for Video Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 408–417.
93. Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; Wei, Y. Deep Feature Flow for Video Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4141–4150.
94. Zhu, M.; Liu, M. Mobile Video Object Detection with Temporally-Aware Feature Maps. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5686–5695.
95. Zhang, C.; Kim, J. Modeling Long- and Short-Term Temporal Context for Video Object Detection. In Proceedings of the 26th IEEE International Conference on Image Processing (ICIP), Tapei, Taiwan, 22–25 Spetember 2019; pp. 71–75.
96. Deng, H.; Hua, Y.; Song, T.; Zhang, Z.; Xue, Z.; Ma, R.; Robertson, N.; Guan, H. Object Guided External Memory Network for Video Object Detection. In Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6678–6687.

97. Beery, S.; Wu, G.; Rathod, V.; Votel, R.; Huang, J. Context R-CNN: Long Term Temporal Context for Per-Camera Object Detection. In Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 13072–13082.
98. Chen, Y.; Cao, Y.; Wang, L. Memory Enhanced Global-Local Aggregation for Video Object Detection. In Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10337–10346.
99. Mao, H.; Yang, X.; Dally, W.J. A Delay Metric for Video Object Detection: What Average Precision Fails to Tell. In Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 573–582.
100. Padilla, R.; Netto, S.L.; da Silva, E.A.B. Metrics for Object Detection. Available online: https://github.com/rafaelpadilla/Object-Detection-Metrics (accessed on 20 December 2020).
101. Computer Research Institute of Montreal (CRIM). thelper Package. Available online: https://thelper.readthedocs.io/en/latest/thelper.optim.html (accessed on 20 December 2020).
102. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. Evaluation Tool for Object Detecttion Metrics. Available online: https://github.com/rafaelpadilla/review_object_detection_metrics (accessed on 20 December 2020).