applied sciences

MDPI

*Review*

# Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review

**Chinthakindi Balaram Murthy [1,†], Mohammad Farukh Hashmi [1,†], Neeraj Dhanraj Bokde [2,†] and Zong Woo Geem [3,*]**

[1] Department of Electronics and Communication Engineering, National Institute of Technology, Warangal 506004, India; balu1602@student.nitw.ac.in (C.B.M.); mdfarukh@nitw.ac.in (M.F.H.)
[2] Department of Engineering - Renewable Energy and Thermodynamics, Aarhus University, 8000 Aarhus, Denmark; neerajdhanraj@eng.au.dk
[3] Department of Energy IT, Gachon University, Seongnam 13120, Korea
[*] Correspondence: geem@gachon.ac.kr; Tel.: +82-31-750-5586
[†] These authors contributed equally to this work.

Received: 15 April 2020; Accepted: 2 May 2020; Published: 8 May 2020

**Abstract:** In recent years there has been remarkable progress in one computer vision application area: object detection. One of the most challenging and fundamental problems in object detection is locating a specific object from the multiple objects present in a scene. Earlier traditional detection methods were used for detecting the objects with the introduction of convolutional neural networks. From 2012 onward, deep learning-based techniques were used for feature extraction, and that led to remarkable breakthroughs in this area. This paper shows a detailed survey on recent advancements and achievements in object detection using various deep learning techniques. Several topics have been included, such as Viola–Jones (VJ), histogram of oriented gradient (HOG), one-shot and two-shot detectors, benchmark datasets, evaluation metrics, speed-up techniques, and current state-of-art object detectors. Detailed discussions on some important applications in object detection areas, including pedestrian detection, crowd detection, and real-time object detection on Gpu-based embedded systems have been presented. At last, we conclude by identifying promising future directions.

**Keywords:** convolutional neural network (CNN); computer vision (CV); graphics processing units (GPUs); object detection; deep learning techniques

## 1. Introduction

Recently, computer vision has been extensively researched in the area of object detection for industrial automation, consumer electronics, medical imaging, military, and video surveillance. It is predicted that the computer vision market will be worth $50 billion by the end of 2020.

For object recognition, the raw input data are represented in matrix pixel form, where the first representation layer abstracts the pixels and encodes edges, the next layer composes and encodes edge arrangement, the next layer up encodes eyes and noses, and the final layer recognizes a face present in the image. Normally, a deep learning process optimally classifies the facial features into their respective levels without supervision.

In object classification application, manual feature extraction is eliminated by a convolutional neural network (CNN), so there is no need to manually identify features that are useful for image classification. CNNs extract features directly from images, and these extracted features are not pre-trained, but they learn while the network is trained on collected images.

Due to automatic feature extraction, deep learning models became highly accurate in computer vision. Deep CNN architecture involves complex models. They require large image datasets for higher accuracy. CNNs require large labeled datasets to perform related tasks in computer vision, such as object classification, detection, object tracking, and recognition.

With the advancement in technology and the availability of powerful graphics processing unit's (GPU), deep learning has been employed on datasets; state-of-the-art results have been demonstrated by researchers in areas such as object classification, detection, and recognition. To perform both training and testing, deep learning requires powerful computational resources and larger datasets. In computer vision, image classification is the most widely researched area and it has attained astonishing results in worldwide competitions through PASCAL, ILSVRC, VOC, and MS-COCO, which apply deep learning techniques [1]. Deep learning techniques are deployed for object detection due to promising results in image classification [2]. Nguyen et al. [3] implemented classification of sonar images with various added noises on GoogleNet CNN and tested on TDI 2017 and 2018 datasets.

In generic object detection, the main aim is to determine whether or not there are any instances of objects from the specified varieties (e.g., animals, vehicles, and pedestrians) in an image, and if present, then return the spatial location and extent of a single object (by bounding box) [4,5]. Object detection became the basis for solving overly complex vision-related tasks; namely, scene understanding, image captioning, instance segmentation, semantic segmentation, object recognition, and tracking [6,7]. The applications of object detection cover areas such as Internet of Things (IoT) and artificial intelligence, which includes intelligent military surveillance systems, security, self-driving cars, robot vision, human–computer interaction (HCI), and consumer electronics.

Recently, deep learning methods [8,9] have emerged as the most powerful techniques for automatically learning features from raw data. Specifically, deep learning methods have achieved great progress in object detection, a problem that has grabbed the attention of many researchers in this decade. Video surveillance is one of the most challenging and fundamental areas in security systems, as it depends entirely on a lot on object detection and tracking. It monitors the behavior of people in public to detect any suspicious behavior [10].

The road-map of object detection milestones is shown in Figure 1.

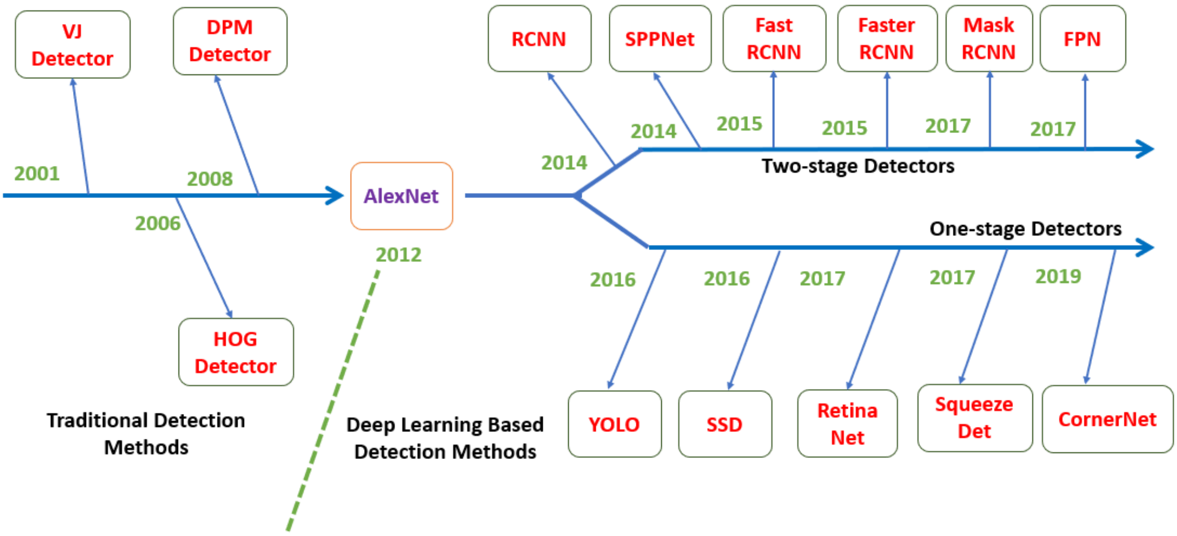

**Figure 1.** Milestones of object detection. In 2012 the major turning point was the use of DCNN implemented for image classification by Krizhevsky et al. [1], VJ Det. [11,12], HOG Det. [13], DPM [14–16], RCNN [17], etc. (source: [18]).

The stated goals of object detection are to achieve both high accuracy and high efficiency by developing robust object detection algorithms.

1. To achieve high detection accuracy, related challenges are:
   - *Intra-class variations*: variations in real-world objects include color, size, shape, material, and pose variations.
   - *Image conditions and unconstrained environments*: factors such as lighting, weather conditions, occlusion, object physical location, viewpoint, clutter, shadow, blur, and motion.
   - *Imaging noise*: factors such as low-resolution images, compression noise, filter distortions.
   - Thousands of structured and unstructured real-world object categories to be distinguished by the detector.
2. To achieve high efficiency, related challenges are:
   - Low-end mobile devices have limited memory, limited speed, and low computational capabilities.
   - Thousands of open-world object classes should be distinguished.
   - Large scale image or video data.
   - Inability to handle previously unseen objects.

### 1.1. Features of the Proposed Review

The proposed survey mainly focuses on providing a thorough and comprehensive review of existing work carried out in deep learning-based object detectors, particularly showing a pathway for new researchers who wish to choose this field.

- Moreover, differently from recently published review papers on object detection topics [18–23], this paper comprehensively reviews modern deep learning-based object detectors starting from regions with convolutional neural netwoks (RCNN) and ending at CornerNet with its pros and cons.
- It also covers some specific problems in computer vision (CV) application areas, such as pedestrian detection, the military, crowd detection, intelligent transportation systems, medical imaging analysis, face detection, object detection in sports videos, and other domains.
- It provides an outlook on the available deep learning frameworks, application program Interface (API) services, and specific datasets used for object detection applications.
- It also puts forth the idea of deploying deep learning models into various embedded platforms for real-time object detection. In the case of a pre-trained model being adopted, replacing the feature extractor with an efficient backbone network would improve the real-time performance of the CNN.
- It describes how a GPU-based CNN object detection framework would improve real-time detection performance on edge devices.

Finally, we intend to give an overview of various deep learning methods deployed on various embedded platforms in real-time objection and possible research directions.

The rest of this paper is organized as follows. Section 2 covers various deep learning architectures used in object detection in detail. Frameworks and API Services, and available datasets and performance metrics for object detection, have been discussed in Sections 3 and 4. Application domains and deep learning approaches for object detection are explained briefly in Sections 5 and 6 respectively. Section 7 discusses various GPU-based embedded systems for real-time object detection implemented using deep learning techniques. Research directions, a conclusion and future research possibilities are presented in Sections 8 and 9.

## 2. Object Detection

### *2.1. Basic Block Diagram of Object Detection*

The intent is to figure out real-world object instances like cats, bicycles, telephones, various flowers, and humans in real-time videos or still images. It paves the way for object recognition, localization, and detection of single/multiple objects within a video frame or an image with a much better interpretation of an image as a whole. Difficult challenges such as occlusion and irregular lighting conditions should be handled carefully while performing object detection. Figure 2 shows the basic block diagram of object detection. The application of object detection covers wide areas, such as medical imaging, security, video surveillance, self-driving vehicles, robot vision, and facial recognition. Figure 3 shows various approaches available in object detection.

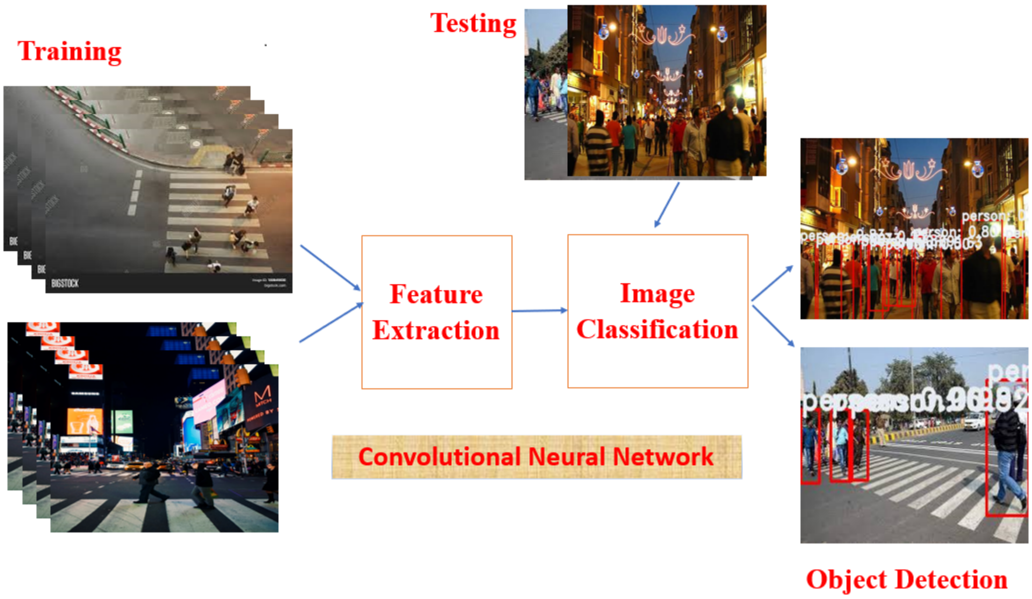

**Figure 2.** Basic block diagram of object detection.

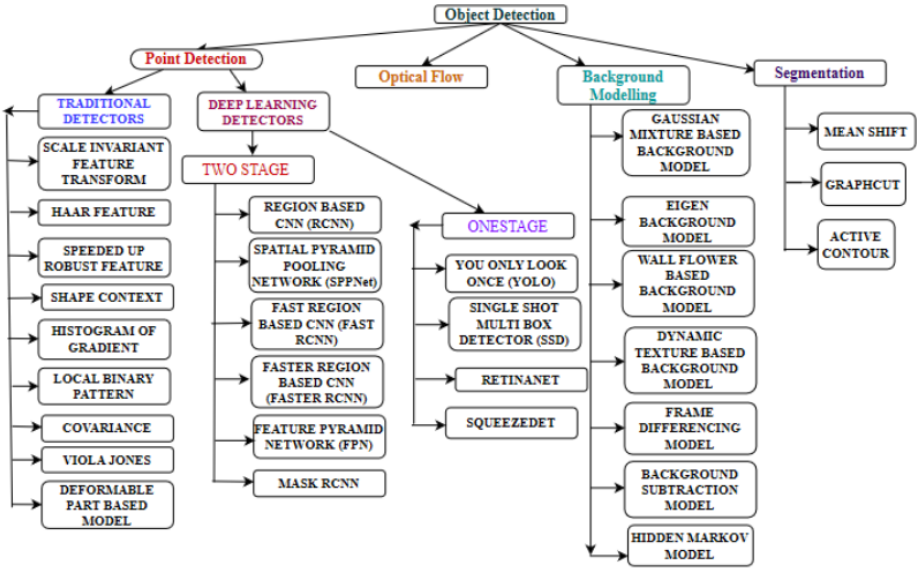
Object Detection
Point Detection
Optical Flow
Background Modelling
Segmentation
TRADITIONAL DETECTORS
DEEP LEARNING DETECTORS
SCALE INVARIANT FEATURE TRANSFORM
HAAR FEATURE
SPEEDED UP ROBUST FEATURE
SHAPE CONTEXT
HISTOGRAM OF GRADIENT
LOCAL BINARY PATTERN
COVARIANCE
VIOLA JONES
DEFORMABLE PART BASED MODEL
TWO STAGE
REGION BASED CNN (RCNN)
SPATIAL PYRAMID POOLING NETWORK (SPPNet)
FAST REGION BASED CNN (FAST RCNN)
FASTER REGION BASED CNN (FASTER RCNN)
FEATURE PYRAMID NETWORK (FPN)
MASK RCNN
ONESTAGE
YOU ONLY LOOK ONCE (YOLO)
SINGLE SHOT MULTI BOX DETECTOR (SSD)
RETINANET
SQUEEZEDET
GAUSSIAN MIXTURE BASED BACKGROUND MODEL
EIGEN BACKGROUND MODEL
WALL FLOWER BASED BACKGROUND MODEL
DYNAMIC TEXTURE BASED BACKGROUND MODEL
FRAME DIFFERENCING MODEL
BACKGROUND SUBTRACTION MODEL
HIDDEN MARKOV MODEL
MEAN SHIFT
GRAPHCUT
ACTIVE CONTOUR

**Figure 3.** Various approaches in object detection.

There are multiple ways to detect objects and these are done using the Viola–Jones (VJ) object detector [11,12], the feature-based object detector [24–26], HOG features using a support vector machine (SVM) classification object detector [13], and object detection-based deep learning techniques. Figure 3 shows various approaches available in object detection.

## *2.2. Various Deep Learning Approaches for Object detection*

### 2.2.1. Viola–Jones Detector

P. Viola and M. Jones performed facial detection without any restrictions (skin color segmentation) [11,12]. The implementation of the VJ detector is simple and straight forward; i.e., a sliding window is moved along all possible locations and the image is scaled; then it checks whether a human face is present in any of the windows. Using the VJ detector, detection speed has improved drastically by including three techniques—integral image, detection cascades, and feature selection.

### 2.2.2. HOG Detector

N. Dalal and B. Triggs [13] first implemented the "histogram of oriented gradients" (HOG) feature descriptor, and it is an improved version of "scale-invariant feature transform" (SIFT) [24,25] and shape contexts [26]. HOG descriptor computes on dense grid cells. To balance both feature invariance (which includes translation and illumination) and linearity (on discriminating different objects classes), overlap local contrast normalization (on blocks) is applied, which improves accuracy. HOG detector is not only used in pedestrian detection but also to detect multiple object classes. Through HOG, the detector resizes the input image multiple times for detecting object sizes, but the size of the detection window is unchanged.

### 2.2.3. Deformable Part-Based Model (DPM)

The deformable part-based model (DPM) was at its peak for traditional object detection methods and was the winner of VOC detection challenges in 2008 and 2009. DPM is an improved version of the HOG detector. Initially, DPM was implemented by P. Felzenszwalb [15], but later R. Girshick [16,17,27,28] made refinements to the DPM detector. The main theme of DPM is "divide and conquer," where the training period is the proper way of learning, and decomposition of objects and the inference period are considered an ensemble of different parts in object detection. A typical DPM detector has two different filters, such as a root-filter and a multiple part-filter. In the DPM detector, all part filter configurations are taught automatically with latent variables using a weakly supervised learning method, instead of specifying the configurations of part filters manually. This process was further refined by R. Girshick as "multi-Instance learning" [29]. Further, improvements in detection accuracy have been obtained by applying techniques, such as "hard negative mining, bounding box regression, and context priming." Girshick sped up the detection technique, achieving 10$x$ faster acceleration than traditional models without the need for sacrificing any accuracy [15,27].

## *2.3. Classification-Based Object Detectors (Two-Stage Detectors)*

Object detection is classified into two groups. They are "one-stage detection" that completes in one step and "two-stage detection," which completes with "coarse to fine" stages.

### 2.3.1. Region-Based Convolutional Neural Network (RCNN)

To overcome the drawbacks proposed by Girshick et al. [17], selecting several regions is eliminated by the RCNN method which uses a selective search method. This method extracts only 2000 regions from the images, and they are also referred to as region proposals. Figure 4 shows RCNN architecture, using a selective search method [30] where a set of object region proposals is extracted. Each object region proposal is transformed into a fixed image size by rescaling it, and then applied to the convolutional neural network model which is pre-trained on ImageNet, i.e., AlexNet [1], for feature extraction. SVM classifier predicts the object presence within each region proposal and also recognizes object classes. RCNN improved mean average precision (mAP) to 58.5% on the VOC-2007 dataset from 33.7% (DPM-v5 [31]).

Despite the great improvement reported by the RCNN method, there are many drawbacks: (1) It consumes more time to train the network, as we need to classify 2000 object region proposals per image. (2) It cannot be implemented in real-time, as each test image needs around 47 s, and since the selective search method is a fixed algorithm, no learning happens at this rate and it leads to the generation of bad object region proposals. To overcome these drawbacks, SPPNet [32] was formulated in the same year.

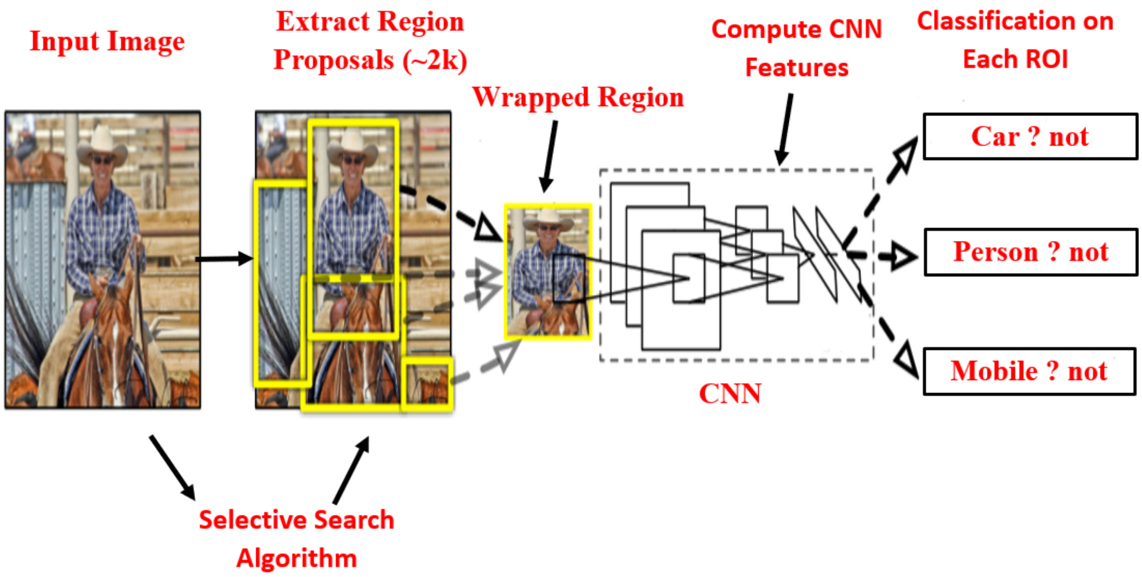
Input Image
Extract Region Proposals (~2k)
Wrapped Region
Compute CNN Features
Classification on Each ROI
Car ? not
Person ? not
Mobile ? not
CNN
Selective Search Algorithm

**Figure 4.** RCNN architecture [17].

### 2.3.2. Spatial Pyramid Pooling Network (SPPNet)

SPPNet [32] was implemented by K.He et al. Figure 5 shows SPPNet architecture. Earlier CNN models require a fixed-size input image; e.g., Alex Net [1] requires an input image of size 224*x*224. SPPNet introduces one "spatial Pyramid Pooling (SPP) layer," which allows the CNN model to produce fixed-length sequence irrespective of the size of region of interest (ROI) or without image resizing. While performing object detection using SPPNet, feature maps are calculated only once from the entire image, and for arbitrary regions, fixed-length sequences are generated using trained detectors, which often avoid computation of the convolutional features. SPPNet performed much faster than RCNN, without losing any detection accuracy, and mAP increased to 59.2% on VOC-2007. Though adequate accuracy is achieved with RCNN, it has many drawbacks: training is still multistaged, it fine-tunes its fully-convolutional (FC) layers, and it ignores earlier layers. To overcome these drawbacks, Fast RCNN [33] was introduced.

### 2.3.3. Fast Region Convolutional Neural Network (Fast RCNN)

Fast RCNN detector [33] was implemented by R. Girshick and is an improvement of SPPNet and RCNN [17,32]. Figure 6 shows Fast RCNN architecture. It allowed us to train simultaneously both detector and bounding box regressor; mAP accuracy increased from 58.5% (RCNN) to 70.0% (Fast RCNN) on the VOC-2007 dataset. All the advantages of RCNN and SPPNet are successfully integrated with Fast RCNN, but still, the detection speed is limited. These drawbacks are eliminated by Faster R-CNN [2].

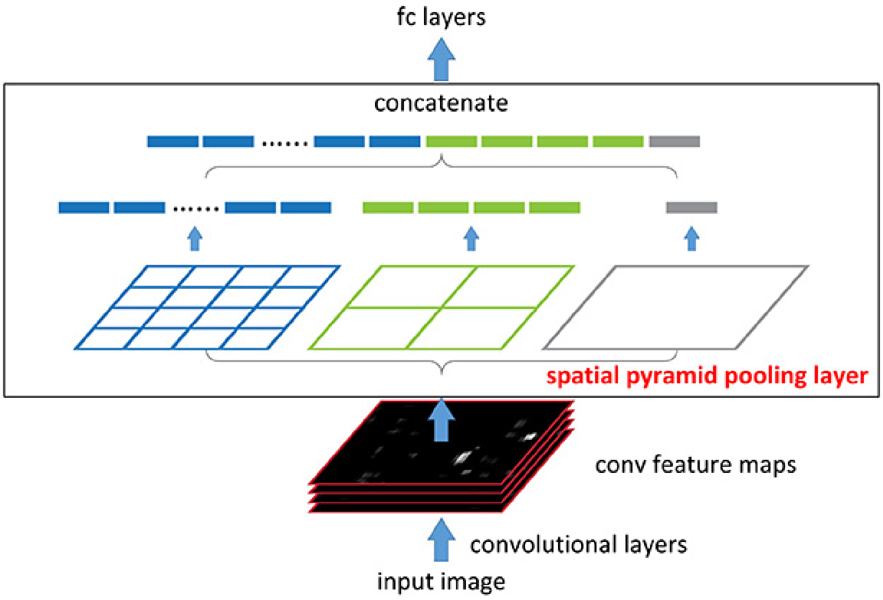

**Figure 5.** Spatial pyramid pooling network architecture [32].

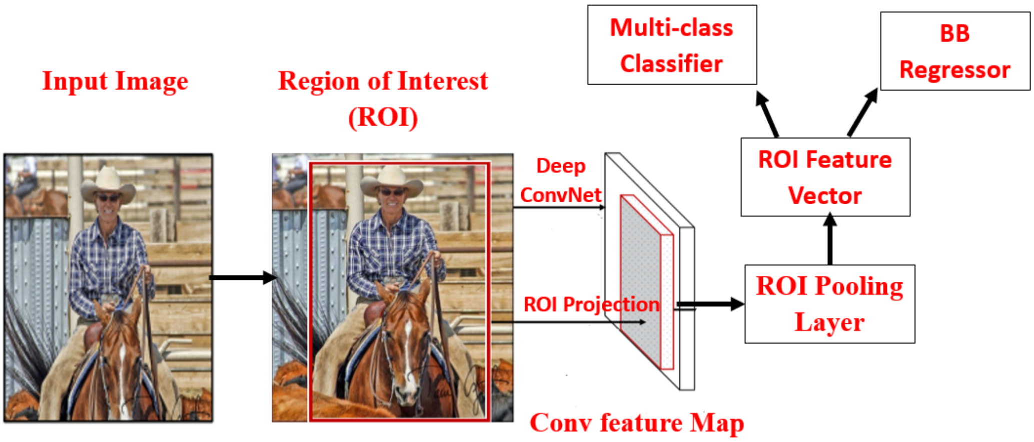

**Figure 6.** Fast RCNN architecture [33].

### 2.3.4. Faster Region Convolutional Neural Network (Faster RCNN)

A Faster RCNN detector [2] was implemented shortly after the introduction of Fast RCNN by S. Ren et al. [2]. To overcome the drawbacks of Fast RCNN, a network referred to as region proposal network (RPN) was introduced in Faster RCNN, as shown in Figure 7. Fast RCNN performs both region proposal generation and detection tasks. Except for RPN, Faster RCNN and Fast RCNN are very similar. Initially, first ROI pooling is performed, and then the pooled area is fed CNN and two FC layers for softmax classification and the bounding box regressor. It is the first near-real-time object detector tested on the MS-COCO dataset; it achieved mAP = 42.7%, VOC-2012, mAP = 70.4%, and 17 fps with ZFNet [34]. Despite Faster RCNN being much faster than Fast RCNN, there is computational redundancy at the final stage. Region-based fully-convolutional networks (RFCN) [35] and Light Head RCNN [36] were further improvements on Faster RCNN.

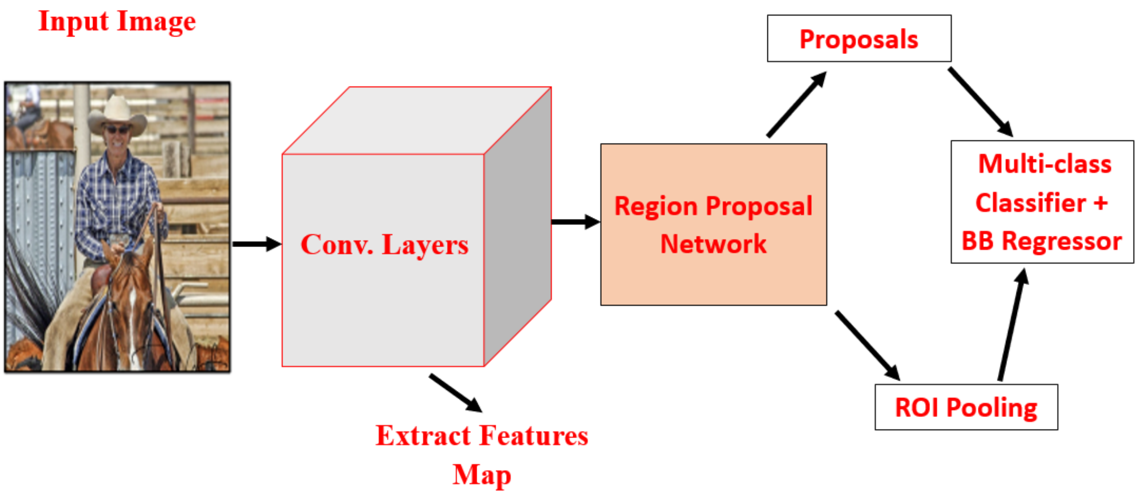

**Figure 7.** Faster RCNN architecture [2].

### 2.3.5. Feature Pyramid Networks (FPN)

Based on Faster RCNN, T-Y. Lin et al. [37] implemented FPN, as shown in Figure 8. Before FPN was introduced, most of the object detectors run detection only at the final layer. For category recognition, the features in CNN deep layers are beneficial but are not conducive for object localization. FPN alone is not a detector, so FPN is implemented along with Fast RCN, Faster RCNN or single shot multi-box detector (SSD). FPN follows top-down pathway architecture and lateral connections while constructing high-level semantics at all scales. It showed great improvement in detecting small objects, since CNN forms a feature pyramid via its forward propagation. FPN + Faster RCNN implementation achieved better detection results on the MS-COCO dataset without any attractive features, i.e., COCO mAP = 59.1%, making it the basic building block for many latest detectors.

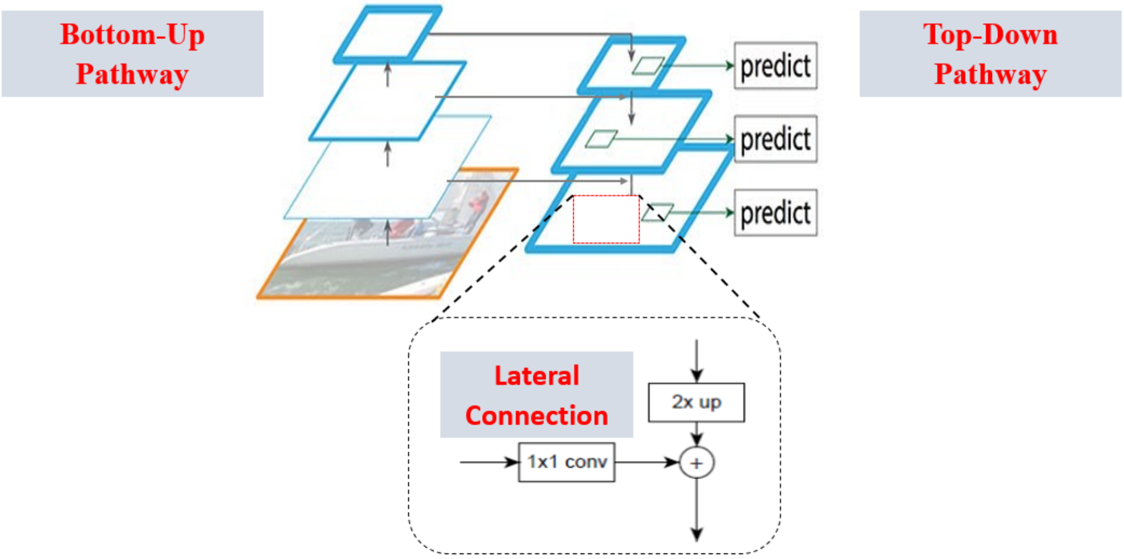

**Figure 8.** Feature pyramid network architecture [37].

### 2.3.6. Mask R-CNN

He et al. [38] introduced Mask R-CNN, and it is the extension of Faster RCNN. Figure 9 shows that Mask R-CNN architecture is a two-stage pipeline. The main aim of Mask RCNN is to solve instance segmentation problems in CV applications; i.e., to separate different objects in an image or a video. Additionally, a mask branch on each region of interest (ROI) is included in Mask R-CNN for predicting an object works in parallel with the class label and bounding box (BB) regression branches. It produces three outputs: a class label, bounding box coordinates, and an object mask. Mask R-CNN efficiently detects objects in the input image or video and concurrently generates a high-quality segmentation mask for each instance detected object. It is conceptually simple to train, flexible, and is a general framework for instance segmentation of objects. To achieve excellent gains in both speed and accuracy, Mask R-CNN uses ResNet-FPN [37,39] as a backbone model for extracting features. But the main drawback is it adds small computational overhead on the network and runs with a speed of nearly 5 Fps.

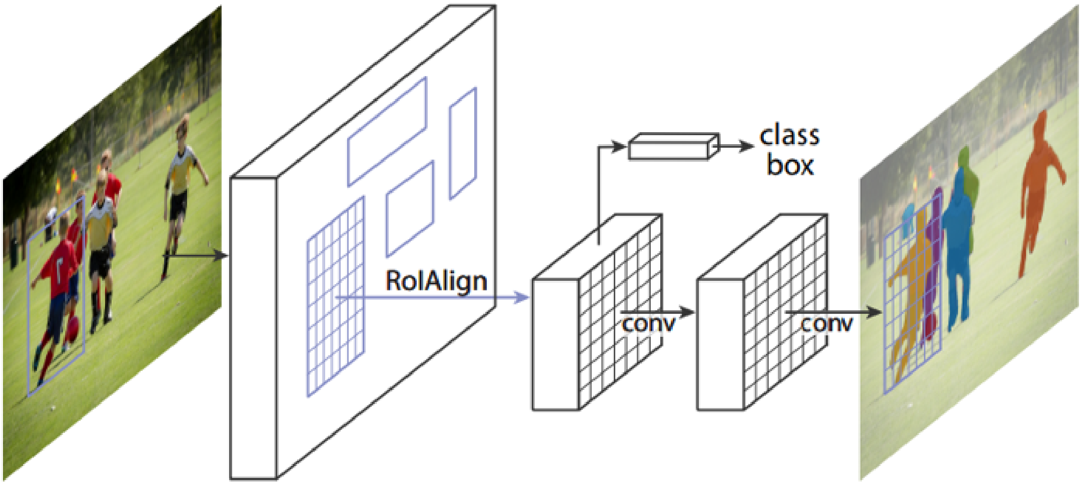

**Figure 9.** Mask R-CNN architecture [38].

## *2.4. Regression-Based Object Detectors (One-Stage Detectors)*

### 2.4.1. You Only Look Once (YOLO)

R. Joseph et al. [40] implemented YOLO architecture, as shown in Figure 10. YOLO is the strongest, fastest, and simplest object detection algorithm used in real-time object detection. YOLO runs at 155 fps achieved $mAP = 52.7\%$, and its improved version runs at 45 fps achieved $mAP = 63.4\%$ on the VOC-2007 dataset. YOLO designers completely replaced the previous object detection model's proposed detection plus verification.

All previous object detection algorithms use regions to localize objects within the image, but the YOLO approach is entirely different; the entire image is applied to a single CNN. YOLO network splits the entire image into regions, and for each region, it predicts bounding boxes and class probabilities. The main drawbacks of the YOLO object detector are: detection of small objects in an image, and localization accuracy dropping off when compared to two-stage detectors. YOLOv2, YOLOv3, and SSD [41] detectors paid much attention to YOLO drawbacks. To the basic YOLO detector, R. Joseph [40] later made improvements and implemented YOLOv2 and YOLOv3 [42,43] which have achieved better detection accuracy without scarifying detection speed.

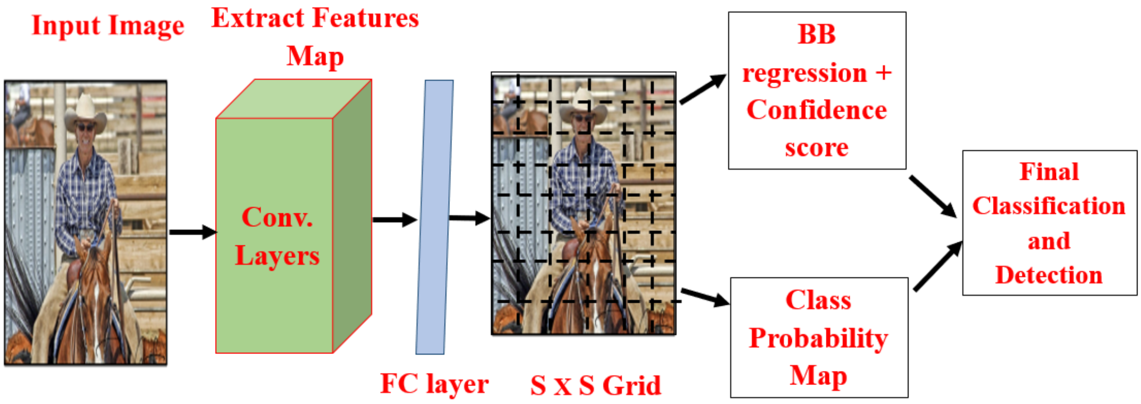

**Figure 10.** You only look once architecture [40].

### 2.4.2. Single Shot Multi-Box Detector (SSD)

W. Liu et al. [41] implemented the single shot multi-box detector (SSD), as shown in Figure 11. SSD is designed purely for real-time object detection in a deep learning era. Instead of taking two shots as in the RCNN series, one for generating region proposals and another for detecting the object of each proposal, it uses only a single shot to detect multiple objects within an image. To improve the detection accuracy of SSD, particularly in detecting small objects, it introduced "multi-reference and multi-resolution detection" techniques. To improve Fast RCNN's real-time speed detection accuracy, SSD eliminated region proposal network (RPN). SSD300 achieves, on the VOC-2012 dataset, mAP = 74.3% at 59 FPS, while SSD500 achieves, on the VOC-2007 dataset, mAP = 76.9% at 22 FPS, which outperforms Faster RCNN (mAP = 73.2% at 7 FPS) and YOLOv1 (mAP = 63.4% at 45 FPS). The drawbacks of SSD: at the cost of speed, accuracy increases with the number of default boundary boxes. SSD detector has more classification errors when compared to RCNN but low localization error while dealing with similar categories.

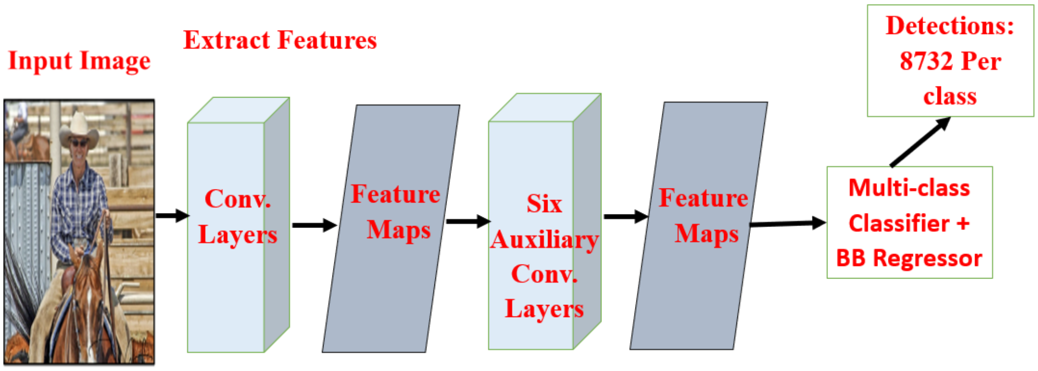
Input Image
Extract Features
Conv. Layers
Feature Maps
Six Auxiliary Conv. Layers
Feature Maps
Multi-class Classifier + BB Regressor
Detections: 8732 Per class

**Figure 11.** Single shot multi-box detector architecture [41].

### 2.4.3. Retina-Net

SSD achieves better accuracy when applied over dense sampling of object locations, aspect ratios, and scales. Large sets of object locations are generated by SSD that densely cover a few areas of the image. This creates a class imbalance as the negatives increase and the object classes present in those locations go undetected. Y. Lin et al. [44] implemented Retina-Net, as shown in Figure 12, to overcome the drawbacks—the class imbalance problem in SSD—and to control the decrease in prediction accuracy of YOLO and SSD. The class imbalance problem in SSD is solved by using focal loss in Retina-Net so that during training, it puts more focus on misclassified examples. Besides maintaining very high-speed detection (MS-COCO dataset 59.1% mAP), focal loss enables SSD to achieve comparable accuracy to that of RCNN series detectors.

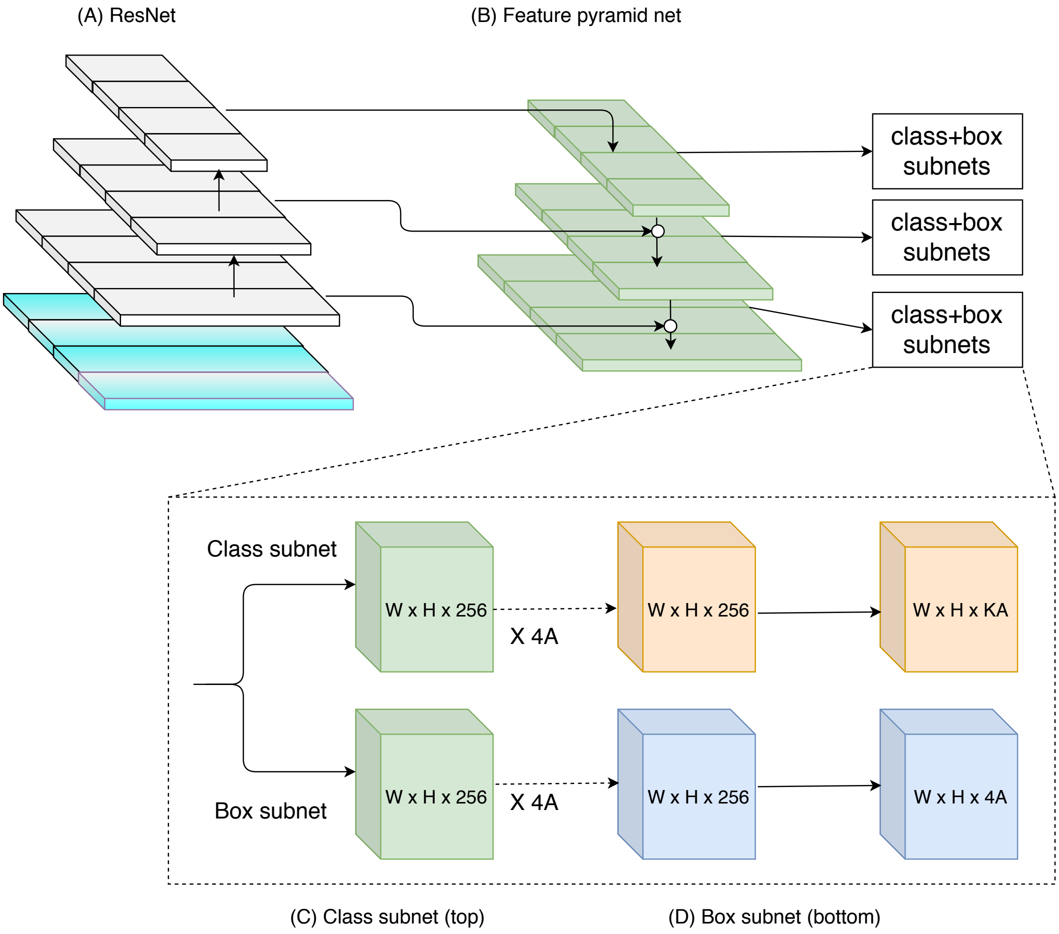

**Figure 12.** Retina-Net architecture [44].

### 2.4.4. SqueezeDet

Wu et al. [45] implemented SqueezeDet, a lightweight, single shot, extremely fast, fully-CNN for detecting objects in an autonomous driving system. To deploy Deep CNN for real-time object detection, the model should address some important problems, such as speed, accuracy, model size, and power efficiency. These constraints are well addressed in the SqueezeDet model, as shown in Figure 13. It is a single forward pass object detector, used to extract a high dimensional, low-resolution feature maps for the applied input images; it uses stacked convolution filters. Second, it uses ConvDet, a convolutional layer fed with a feature map as input that produces a large number of bounding boxes and also predicts the object's category. Finally, by applying filtering to these bounding boxes, it outputs final object detections. The backbone model of SqueezeDet is SqueezeNet [46], and the model size is less than 8 MB which is very small compared to AlexNet [1] without losing any accuracy. This model consists of approximately two million trainable parameters and achieves a higher level of accuracy when compared to VGG19 and ResNet-50 with 143,000,000 and 25,000,000 parameters. For the input image of size 1242x375, this model achieved 57.2 Fps on the Kitti dataset [47] and consumed only 1.4 J energy per image.

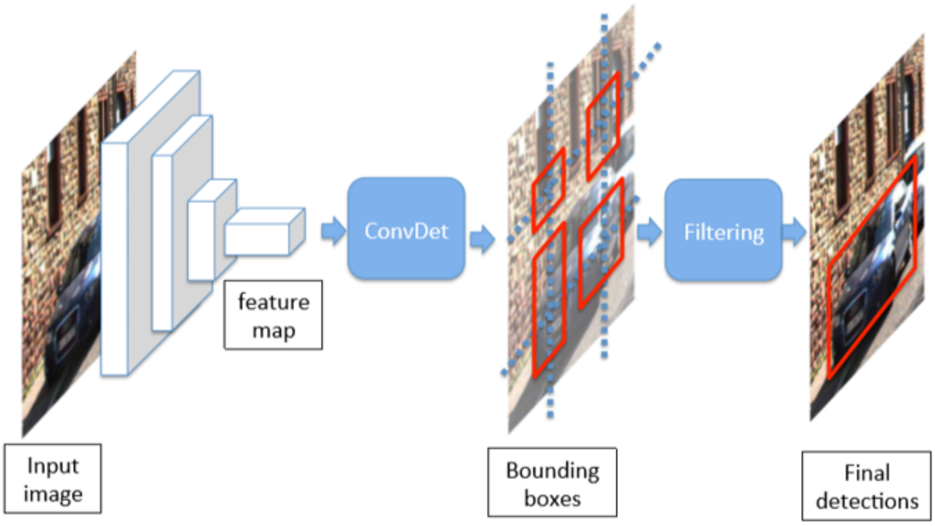

**Figure 13.** SqueezeDet architecture [45].

### 2.4.5. CornerNet

Law et al. [48] implemented CornerNet for object detection, wherein the object is detected by a pair of key points using a CNN instead of drawing an anchor box around the detected object. So the need for designing anchor boxes usually used in one stage detectors is eliminated by detecting objects as paired key points; i.e., top-left and top-right corners respectively. They introduced a new type of pooling layer referred to as corner pooling, which helps the network to localize corners better. The CNN outputs the heatmap for all top-left corners and bottom-right corners, along with an embedded vector map for each detected corner. On the MS-COCO dataset, CornerNet achieved 42.2% AP which outperforms the existing one-stage detectors. Figure 14 shows the CornerNet architecture. The main drawback is that it generates incorrect paired key points for the detected object. So to overcome this drawback, Duan et al. [49] implemented CenterNet by introducing a third key point at the center to detect each object. CenterNet achieved 47% AP, and inference speed is slower than CornerNet. Table 1 shows the summary of different object detection algorithms tested on Pascal Titan X GPU on MS-COCO and Pascal-VOC2007 datasets. Table 2 shows a comparison of various deep learning-based object detectors' performances on the MS-COCO test-dev dataset.

**Table 1.** Summary of different object detection (Pascal Titan X GPU) performances on MS-COCO and Pascal-voc07.

| S.No | Architecture | mAP (MS-COCO) | mAP (Pascal-Voc 2007) | FPS |
|---|---|---|---|---|
| 1 | RCNN [17] | – | 66% | 0.1 |
| 2 | SPPNet [32] | – | 63.10% | 1 |
| 3 | Fast RCNN [33] | 35.90% | 70.00% | 0.5 |
| 4 | Faster RCNN [2] | 36.20% | 73.20% | 6 |
| 5 | Mask RCNN [44] | - | 78.20% | 5 |
| 6 | YOLO [40] | – | 63.40% | 45 |
| 7 | SSD [41] | 31.20% | 76.80% | 8 |
| 8 | YOLOv2 [42] | 21.60% | 78.60% | 67 |
| 9 | YOLOv3 [43] | 33.00% | – | 35 |
| 10 | SqueezeDet [45] | - | – | 57.2 |
| 11 | SqueezeDet+ [45] | - | – | 32.1 |
| 12 | CornerNet [48] | 69.2 | – | 4 |

**Table 2.** Comparison of deep learning based object detection performances on MS-COCO test-dev dataset. (Note: SqueezeDet* and SqueezeDet+* models are trained on Kitti dataset. AP(E), AP(M), and AP(H) refer to average precision for easy, medium, and hard cases.)

| Architecture | Backbone Model | AP | AP(E) | AP(M) | AP(H) |
|---|---|---|---|---|---|
| | Two-stage Detectors | | | | |
| RCNN [17] | VGG16 | – | – | – | |
| SppNet [32] | VGG16 | – | – | – | – |
| Fast RCNN [33] | VGG16 | 19.7 | – | – | – |
| Faster RCNN with FPN [37] | VGG16 | 36.2 | 18.2 | 39 | 48.2 |
| Mask RCNN [44] | ResNext-101 | 39.8 | 22.1 | 43.2 | 51.2 |
| | One-stage Detectors | | | | |
| YOLOv2 [42] | DarkNet53 | 33 | 18.3 | 35.4 | 41.9 |
| YOLOv3 [43] | DarkNet19 | 21.6 | 5 | 22.4 | 35.5 |
| SSD300 [41] | VGG16 | 25.1 | 6.6 | 24.4 | 36.5 |
| SSD512 [41] | VGG16 | 28.8 | 10.9 | 31.8 | 43.5 |
| SSD513 [50] | ResNet-101 | 31.2 | 10.2 | 34.5 | 49.8 |
| RetinaNet500 [40] | ResNet-101 | 34.4 | 14.7 | 38.5 | 49.1 |
| RetinaNet800 [40] | ResNet101-FPN | 39.1 | 21.8 | 42.7 | 50.2 |
| SqueezeDet* [45] | SqueezeNet | 76.7 | 77.1 | 68.3 | 65.8 |
| SqueezeDet+* [45] | SqueezeNet | 80.4 | 81.4 | 71.3 | 68.5 |
| CornerNet511(single-scale) [46] | Hourglass-104 | 40.6 | 19.1 | 42.8 | 54.3 |
| CornerNet511(multi-scale) [46] | Hourglass-104 | 42.2 | 20.7 | 44.8 | 56.6 |

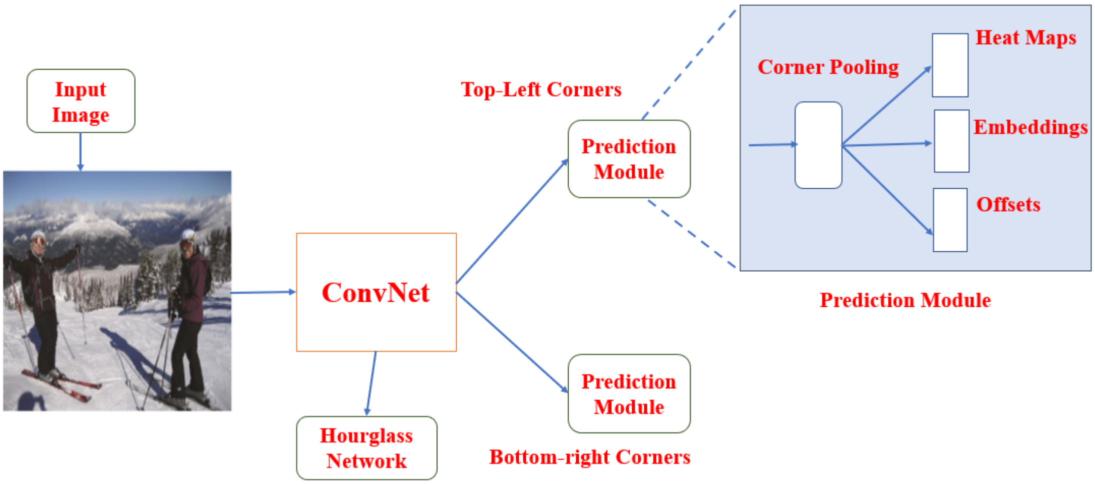

**Figure 14.** CornerNet architecture [48].

## 3. Available Deep Learning Frameworks and API Services

Table 3 shows various deep learning frameworks. Deep learning frameworks are described in the form of framework designer; features exhibited; supported platforms; languages; models supported, such as CNN, RCNN, and DBN/RBM; parallel execution; and license. Table 4 shows the list of available API services for object detection.

**Table 3.** Comparison of various deep learning frameworks.

| Name | Designer | Software License | Supported Platforms | Features | Languages Supported | RNN | CNN | DBN RBM | Parallel Execution |
|---|---|---|---|---|---|---|---|---|---|
| Wolfram Mathematica [51] | Wolfarm Research | Proprietary | Windows, macOS, Linux, Cloud computing | Machine learning, data science, image processing, neural networks, geometry, visualization | C++, Wolfram Language, CUDA | Yes | Yes | Yes | Yes |
| Dlib [52] | Davis King | Boost software license | Cross Platform | Used for creating robust and complex software in C++ to solve real-world problems. Useful for both industry and academia | C++ | No | Yes | Yes | Yes |
| Theano [53] | Université de Montréal | BSD | Cross Platform | Use GPUs and perform symbolic differentiation, to define, optimize and evaluate expressions involving multi-dimensional arrays efficiently | Python | Yes | Yes | Yes | Yes |
| Caffe [54] | Berkeley Vision and Learning Center | BSD | Linux, macOS, Windows | Expressive architecture and speed. By setting a single flag, switches from CPU to GPU, to train on a GPU machine and deploy it on handheld devices. | Python, MATLAB, C++ | Yes | Yes | No | X |
| Deeplearning4j [55] | Deeplearning4j community; originally A. Gibson | Apache 2.0 | Windows, macOS, Linux, Android | It combines variational autoencoders, sequence-to-sequences autoencoders, convolutional nets or recurrent nets as needed in a distributed deep learning framework | Java, Scala, Clojure, Python (Keras) | Yes | Yes | Yes | Yes |
| Chainer [56] | Preferred networks | BSD | Linux, macOS | Supports GPU acceleration using CUDA, Supports higher-order derivatives, easy to use APIs. | Python | Yes | Yes | No | Yes |
| Keras [57] | Francois Chollet | MIT License | Linux, macOS, Windows | Fast experimentation with DNN's, modular and extensible. It allows distributed training of DNN models on clusters of GPUs and Tensor processing units (TPUs). | Python, R | Yes | Yes | No | Yes |
| MATLAB+ Deep learning Toolbox [58] | Mathworks | Proprietary | Linux, macOS, Windows | MATLAB supports interoperability with open-source deep learning frameworks using ONNX import and export capabilities. Preprocess datasets fast with domain specific apps for audio, video, and image data. | Matlab | Yes | Yes | No | Required Parallel Computing Toolbox |
| Apache Singa [59] | Apache Incubator | Apache 2.0 | Linux, macOS, Windows | It provides a flexible architecture for scalable distributed training and is extensible to run over a wide range of hardware | C++, Python, Java | Yes | Yes | Yes | Yes |

**Table 3.** *Cont.*

| Name | Designer | Software License | Supported Platforms | Features | Languages Supported | RNN | CNN | DBN RBM | Parallel Execution |
|---|---|---|---|---|---|---|---|---|---|
| Tensorflow [60] | Google Brain | Apache 2.0 | Windows, macOS, Linux, Android | Build and train ML models easily using intuitive high-level APIs like Keras, automated image captioning software, flexible architecture, computations are expressed as stateful dataflow graphs | Python (Keras), C/C++, Java, R, Julia, Swift | Yes | Yes | Yes | Yes |
| PyTorch [61] | Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan | BSD | Linux, macOS, Windows | Tensor computing with strong acceleration via GPU, DNN's built on a tape-based auto diff system, Useful for applications such as deep learning and NLP | C++, Python | Yes | Yes | | Yes |
| BigDL [62] | Jason Dai | Apache 2.0 | Apache Spark | distributed deep learning framework for Apache Spark | Scala, Python | Yes | Yes | No | X |
| Neon [63] | Intel Nervana | Apache 2.0 | Apache Spark | Supports automatic differentiation, Fast performance on various hardware & DNN's (GoogLeNet, VGG, AlexNet, GAN's), Support for commonly used models including convnets, RNNs, LSTMs, and autoencoders. | Python | Yes | Yes | Yes | Yes |
| Apache MXnet [64] | Apache Software Foundation | Apache 2.0 | Linux, macOS, Windows, AWS, Android, iOS, JavaScript | A merger of symbolic and imperative programming, auto-differentiation, portability. | Matlab, C++, Go, Python, JavaScript, Julia. | Yes | Yes | Yes | Yes |

**Table 4.** List of available API services for object detection.

| Name | Services | Features | Accessibility |
|---|---|---|---|
| Microsoft Cognitive Service [65] | computer vision | Face, vision and speech recognition,<br>object motion tracking,<br>facial expression recognition,<br>speech understanding, image tagging,<br>language understanding. | REST API |
| Amazon Rekognition [66] | image recognition | Activity, scene and object detection,<br>facial recognition on images and video, text in images,<br>detect unsafe video. | HTTP |
| IBM Watson Vision Recognition Service [67] | understanding content in the images | Train a custom model for visual inspection, face detection, image class description and taxonomy,<br>image matching identification and,<br>supports multi-languages. | HTTP |
| Google Cloud Vision [68] | image analysis | Content detection, face, landmark and logo detection, image sentiment analysis. | Integrated REST API |
| Cloud Sight [69] | image understanding | Via REST API when an image is forwarded, then the response is given as image Description. | REST API |

## 4. Object Detection Datasets and Metrics

Since all CV algorithms are trained and verified using datasets only, datasets play a crucial role in achieving correct outputs.

- ImageNet [70]: It is based on WordNet Hierarchy. WordNet is also referred to as Synset. To define each synset, on average 1000 images are provided by ImageNet. ImageNet dataset offers billions of images in WordNet Hierarchy. It has a total of 14,197,122 images—images with bounding box annotations of 1,034,908 and an image resolution of 480*x*410 pixels. In ImageNet, to detect local features, 1.2 million images exhibit SIFT (scale-invariant feature transform) features.
- WIDERFACE [71]: This dataset contains total of 32,202 images and which includes around 400,000 faces for a wide range of scales. The dataset is split into three parts: training data 40%, validation data 10%, and testing data 50%.
- FDDB [72]: "Face Detection Dataset and Benchmark" contains 2845 images with a total of 5171 faces. Since it is a small dataset, it is generally used for testing only, and WIDERFACE dataset is used for training the object detector.
- CityPersons [73]: It is a newly created and challenging pedestrian dataset on the top of the Cityscapes [74] dataset. This dataset is very useful, especially in more difficult cases, such as with small-scale data and heavily occluded pedestrians. It contains 5000 images that were captured from various cities in Germany.
- INRIA [13]: INRIA is a popular person dataset used in pedestrian detection. It contains 614 person images for training and 288 person images for testing.
- KITTI [47]: This dataset contains high resolution 7481 labeled images and 7518 testing images. "Person class" in this dataset is divided into two subclasses—pedestrian and cyclist, and it is tested on three evaluation metrics—easy (E), moderate (M) and hard (H).
- ETH [75]: This dataset contains three videoclips and which have a total of 1804 frames, and it is commonly used as a testing dataset.
- 80M Tiny image dataset [76]: This dataset contains around 80 million 32*x*32 colored images.
- Microsoft COCO (MS-COCO) [77]: "Microsoft Common Objects in Context" dataset has 330,000 images comprising 250,000 labeled images; 150 are object instances, 80 are object categories, 91 are stuff categories, and 5 are captions per image. This dataset exhibits features such as context recognition, multi-objects per image, and object segmentation.

- CIFAR-100 dataset [78]: This dataset provides only 100 object classes; each contains 600 images of which 500 and 100 per class are training and testing images; 100 object classes are clustered into 20 superclasses, and each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs).
- CUB-200-2011 [79]: Cub-200 dataset [80] consists of 200 bird species annotated classes, and each class has 11,788 images. Every annotated image has a single bounding box image (annotations per image: 15 part locations, 312 binary attributes, and one bounding box).
- Caltech-256 [81]: It contains 256 object classes with a total of 30,607 images for each class 80 images, and is not suitable for object localization.
- ILSVRC [82]: Since 2010, the "ImageNet Large Scale Visual Recognition Challenge (ILSVRC)" has been conducted every year for object detection and classification. The ILSVRC dataset contains 10 times more object classes than PASCAL VOC. It contains 200 object classes, whereas the PASCAL VOC dataset contains only 20 object classes.
- PASCAL VOC [83]: The "Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes (VOC)" challenge provides standardized image datasets for object class recognition tasks, and a common set of tools to access available datasets, and enables evaluations and comparisons of the performances of various object detection methods from 2008 to 2012. For object detection, all the researchers mostly follow MS-COCO and PASCAL-VOC datasets.

Table 5 gives brief summary of the main stages of PASCAL VOC developments and image dataset statistics [4], and includes new challenges every year.

**Table 5.** Description of VOC dataset challenges.

| Year | Classes | Images | Annotated Objects | Segmentation | New Developments in Addition to Classification and Detection Challenges |
|---|---|---|---|---|---|
| 2008 | 20 | 4340 | 10,363 | nil | occlusion flag added to annotations |
| 2009 | 20 | 7054 | 17,218 | 3211 | improved dataset along with new augmented images |
| 2010 | 20 | 10,103 | 23,374 | 4203 | classification action introduced based on ImageNet |
| 2011 | 20 | 11,530 | 27,450 | 5034 | classification action extended to 10 classes plus others |
| 2012 | 20 | 11,530 | 27,450 | 6929 | size of Segmentation dataset substantially increased, annotated with a reference point on the body |

Table 6 shows image classification dataset challenges. Table 7 shows a comparison of object detection datasets between PASCAL VOC and ILSRVC. Some datasets like LabelMe [84], SUN (Scene-understanding) 2012 [85] provide better "image annotation than image labeling." LabelMe contains 187k images; by using bounding polygon all objects are annotated. SUN dataset [85] has 900 classes of scenes, and more than 135K images. Compared to ImageNet, Open Images [86] is much larger and contains more than 9 million real-life images with 6000 classes. For autonomous driving, a specific dataset KITTI vision [47] is used and the images are taken from the mid-size city. For agriculture application FieldSAFE [87] dataset is used and it holds around 2 h of raw sensor data collected on a grass moving scenario.

The following are the standard metrics used for evaluating the performance of detection algorithms: average precision (AP), mean average precision (mAP), speed (frames per second), true positive, false positive, IOU threshold, recall, and confidence threshold. Mean average precision (mAP) is widely used as a performance evaluation metric for object detection.

**Table 6.** ILSVRC image classification dataset statistics.

| Year | Main Challenges |
|---|---|
| 2012 | Fine-grained Classification, Classification, Classification, and localization |
| 2013 | Detection, Classification, Classification with localization |
| 2014 | Detection, Classification, and localization |
| 2015 | Object localization, Scene Classification, Object detection from video |
| 2016 | Object localization, object detection from video, Scene Classification and parsing |
| 2017 | Object localization, Object detection, Object detection from video |

**Table 7.** Comparison of object detection datasets between PASCAL VOC and ILSRVC.

| | | PASCAL VOC 2012 | ILSRVC 2013 | ILSRVC 2014 |
|---|---|---|---|---|
| **Number of Object Classes** | | **20** | **200** | **200** |
| Training | No. of images | 5717 | 395909 | 456567 |
| | No. of objects | 13609 | 345854 | 478807 |
| Validation | No. of images | 5823 | 20121 | 20121 |
| | No. of objects | 13841 | 55502 | 55502 |
| Testing | No. of images | 10991 | 40152 | 40152 |
| | No. of objects | _ | _ | _ |

## 5. Object Detection Application Domains

Object detection is applied in wide areas of CV, including defense (surveillance), iris recognition, face detection, human-computer interaction (HCI), robot vision, security, medical imaging, smart transportation, automated vehicle systems, image retrieval system, and machine inspection. For continuous video surveillance for a few hours, sensors generate petabytes of image data. The generated data is further reduced to geospatial data and then integrated with the other collected data to get a clear-cut picture of the current situation. One of the most important tasks involving object detection is to track vehicles/suspicious people from the collected raw data [88]. Crucial applications of object detection include detecting faulty electric wires, detecting unattended baggage, detecting driver drowsiness on highways, detecting vehicles parked in restricted areas, detecting objects present or coming onto the road (for self-driving vehicles), and also detecting stray animals present in industrial areas.

All the above requirements for applications may vary according to the circumstance, and detection is performed either offline or online. Factors such as inter-class and intra-class variations, occlusions, rotation invariance, and multi-pose are also the main challenges in object detection.

### *5.1. Pedestrian Detection*

An important application area of object detection is pedestrian detection. It is used extensively in complex applications, including video surveillance, self-driving cars, etc. Earlier pedestrian detection methods used for object detection, such as the HOG detector [13] and the integral channel features (ICF) detector [89], rely purely on terms of feature representation [13,89], classifier design [90], and acceleration detection [91]; others, such as "detection by components" [18,92–94], gradient-based representation [16,95,96], and the deformable part-based model (DPM) [17,27,79] are used for pedestrian detection.

In pedestrian detection the main difficulties and challenges faced are (a) small pedestrian detection; (b) hard negatives; (c) real-time pedestrian detection from HD video; (d) dense and occluded pedestrians.

The first CV task that applied deep learning was pedestrian detection [97]. The recently improved Tiny-YOLOv3 [98] implementing pedestrian detection on the Pascal-Voc 07 dataset achieved better accuracy compared to Tiny-YOLOv3. Table 8 shows possible directions to overcome the major challenges and difficulties faced in pedestrian detection. Table 9 shows various papers applied deep learning-based techniques to handle dense and occluded pedestrian detection.

The widely used datasets for evaluating the pedestrian detection performance are PASCAL-VOC [83], INRIA [13], KITTI [47], CalTech [81], ETH [75], and CityPersons [73].

**Table 8.** Remedies to improve challenges arising in pedestrian detection.

| Challenges | Method |
|---|---|
| To improve accuracy of small pedestrian detection | Feature fusion [99]<br>Integral feature pyramid [37]<br>Topological line localization [100]<br>High-resolution handcrafted features [101–103]<br>Ensemble detection [104]<br>Feature correlation layer [105]<br>Cascaded detection [106]<br>'Visual attention mechanism called as<br>Region Context Network (RCN)' [107] |
| To improve dense and occluded detection | 'Ensemble of part detectors' [108,109]<br>'Guided attention mechanism' [110]<br>'Adaptive zoom-in' techniques [111,112]<br>Designing new loss function by considering both the attraction of target and repulsion of surrounding objects. [113] |
| To improve hard negative detection | By integration of 'Boosted forest [100] and semantic segmentation' [114]<br>'Bootstrap' [41,99,115,116]<br>'Anchor refinement module introduced in RefineDet'. [117]<br>Designing new loss functions [40,118,119].<br>'Cross-modal learning' used to enrich the features of hard negatives using both RGB and infrared images [120] |

**Table 9.** Various techniques to handle occlusion situations while detecting pedestrians.

| Articles | Methods | Datasets | Remarks |
|---|---|---|---|
| Tian Y et al. [108] | CNN | Caltech | Proposed Deep parts CNN detector is trained on weakly labelled data and can detect pedestrian by observing only a part of a proposal. |
| Ouyang et al. [109] | Deep CNN | Caltech | 'Feature extraction, deformation handling, occlusion handling, and classification are four important components in pedestrian detection uses deep CNN jointly learned' in order to maximize their strengths through cooperation. |
| Zhang S et al. [121] | occlusion-aware R-CNN | CityPersons, ETH, INRIA and Caltech | 'Used a new part occlusion-aware region of interest ETH, INRIA (PORoI) pooling unit in place of RoI pooling layer and Caltech in order to integrate the prior structure information of human body with visibility prediction into the network to handle occlusion.' |
| Zhou C et al. [122] | CNN | Caltech and CityPersons | 'Bi-box Regression for Pedestrian Detection and Occlusion Estimation.' |
| Hsu W Y et al. [123] | multiscale block-based HOG's via Gabor filtering | Caltech | Method effectively processes images in which a crowd is present or pedestrians are partially occluded and enables pedestrian detection in images of different scenes. |
| Ren Y et al. [124] | Deformable Faster RCNN | SORSI and HRRS | 'A deformable Faster R-CNN is constructed by substituting the standard convolution layer with a deformable convolution layer in the last network stage' for occluded object detection. |
| Li W et al. [125] | Enhanced Cascade detector | INRIA | Proposed improved adaptive boosting (Adaboost) algorithm and enhanced cascade detector output to detect partially occluded pedestrians. |

### *5.2. Face Detection*

The oldest CV application is face detection. Earlier, the VJ detector [12] was used for face detection and is still playing a vital role in today's object detection. Face detection is used in everyday life, including in the attendance monitoring of students, smartphone authentication, and facial recognition for criminal investigations.

In face detection, the main difficulties and challenges faced are: (a) occlusion; (b) multi-scale detection; (c) intra-class variation; (d) real-time detection. Face detection research history is divided into two time slots: traditional and deep learning periods. From 2000 to 2015, it was a traditional face detection period, and thereafter, it was a deep learning face-detection era.

In traditional face detection methods [126–134] were constructed based on:

- "Boosted decision trees" [11,12,135], as they are easy to compute, but for complex scenes they provide only low detection accuracy.
- CNNs which are used to speed-up detection where the computation of features is shared [136,137].

Deep learning-based face detection methods implement deep learning algorithms such as Faster RCNN and SSD. Table 10 describes possible directions to overcome the major challenges and difficulties faced in face detection.

**Table 10.** Remedies to improve challenges arising in face detection.

| Challenges | Methods |
|---|---|
| To improve speed up face detection | Cascaded detection [138,138,139]<br>'To predict the scale distribution of the faces in an image and then run algorithm on some selected scales' [42,140,141] |
| To improve multi-pose face detection | 'Face calibration method' using progressive calibration through multiple detection stages [142]<br>Estimating calibration parameters [143] |
| To improve accuracy of occluded faces | 'Detection based on parts' [144,145]<br>'Attention mechanism' which highlights underlying face target features [146]<br>GAN is used for improving occluded objects by applying adversarial training which generate occlusion masks. [147] |
| To improve multi-scale face detection | improved by using similar detection strategies as that of generic object detection [148–150]<br>'Multi-scale feature fusion' [37,41,50,117,151]<br>'Vote-based ensemble' method<br>'Multi-resolution detection' [152]<br>(dilated convolution) [18,41,153–155] |

The widely used datasets for evaluating face detection performance are WIDERFACE [71], PASCAL-VOC [5], and Face Detection Dataset and Benchmark (FDDB) [72].

### *5.3. Military Applications*

One of the major applications of the military field covers the areas of remote sensing and flying-object detection. Remote sensing detection aims at detecting objects in remote areas. There are high resolution input images, but for practical use, small objects make the current detection procedure too slow, and due to complex backgrounds, it can cause serious misdetections.

The data fusion technique was adopted by the researchers to solve these challenges. Captured images are located far away from a viewpoint, so strong pipeline CNN models are required, such as RetinaNet, SSD, YOLO, and SqueezeNet—these are difficult to adapt to new domains. So designing new remote sensing detectors and remote sensing datasets remains still a hot research topic.

Zhang et al. [156] implemented a ”weakly supervised learning framework based on coupled CNN” for aircraft detection. Han et al. [157] proposed a “weakly supervised and high-level feature learning” framework for detecting optical remote sensing images. Li et al. [158] implemented a novel cascaded DNN architecture by combining Hough transform and multi-stage RPN for dense detection of buildings in remote sensing images. Mou et al. [159] proposed to a multitask learning network that performs two tasks concurrently: segmenting vehicles and detecting semantic boundaries. A hybrid DNN proposed in [160] exacts only the same scale features for detecting small targets; i.e., vehicles in satellite images. A deep learning method starts by segmenting the input image into small homogenous regions which are used for detecting cars in UAV images [161].

Ma et al. [162] proposed a “multi-model decision feature network which takes both contextual information and multi-region features” for detecting objects in remote sensing images. A double multi-scale FPN which consists of a multi-scale RPN and multi-scale object detection network on very high resolution (VHR) remote sensing imagery, especially for small and dense objects [163]. Semantic attention DEEP neural networks separate objects of interest and cluttered background while detecting objects and improve detection accuracy in aerial images [164]. Cheng et al. [165] implemented “rotation invariant CNN to deal with the problem of object rotation variations” in optical remote sensing images.

Li et al. [166] implemented “R3–Net by combining rotatable RPN and rotatable detection network for multi-oriented vehicle detection” in aerial images or videos. Tang et al. [116] implemented “improved Faster RCNN [which] aims at reducing false vehicle detection by negative example mining.” “$R^2$–CNN real-time tiny object detection” was implemented for large-scale remote sensing images [167]. A “deep CNN model based on improved bounding box regression and multi-level feature fusion” was implemented for detecting objects in remote sensing images [168].

Typical datasets used for evaluating remote sensing object detection performance are NWPU VHR [169], DLR 3K Munich [170], VEDAI [171], DOTA [47], and HRRSD [172].

### *5.4. Medical Image Analysis*

In medical image analysis specifically, tumor detection, skin disease detection, tumor segmentation, brain image analysis, glaucoma detection, healthcare monitoring, etc., are the major fields to apply deep learning-based techniques to. In the medical field, detection is commonly referred to as “computer-aided detection (CAD).” CAD systems aim to detect the abnormalities in a patient as early as possibly; for instance, in cases of breast cancer or lung cancer.

Islam et al. [173] implemented “two-deep CNN networks Inception v4 [174] and ResNet [39] to detect Alzheimer’s disease using MRI images” of OASIS dataset [175]. The “Deep CNN model [was] proposed for epilepsy lesion detection in multiparametric MRI images using auto-encoders” [176]. Laukampet et al. [177] proposed a “multi-parametric deep CNN model for meningiomas detection” in the brain and a “deep CNN technique for the estimation of colorectal cancer in CT tumor images” for early treatment [178].

Bejnordi et al. [179] proposed “deep CNN techniques for metastases detection in hematoxylin tissue sections of the lymph nodes” subjected to cancer. A “fully-convolutional network-based heat regression method [was] implemented for the detection of breast mass” in mammography images

[180]. A “CAD system based on a deep CNN model to detect breast cancer” in MRI images was proposed in [181].

Abramoff et al. [182] proposed the “CNN technique to detect diabetic retinopathy in fundus images” using public datasets. A “3D group-equivariant CNN technique for lung nodule detection” in CT images was proposed in [183]. Recently, deep learning-based techniques have been used for diagnosing retinal diseases [184,185]. Li et al. [177] introduced a CNN-based attention mechanism for glaucoma detection. A deep CNN model [186] was introduced for melanoma detection, and there was also a “Deep CNN for the detection of chronic obstructive pulmonary disease (COPD) and acute respiratory disease (ARD) prediction” in CT images of smokers [187].

The widely used datasets for evaluating medical image analysis performance are different for different diseases. ILD [188] and LIDC-IDRI [189] datasets for the lung; ADNI [190], BRATS [191], and OASIS-3 [175,192] datasets for the brain; DDSM [193], MIAS [194], CAMELYON 17 [195], and INbreast [196] datasets for the breast; DRIVE [197], STARE [198], and MESSIDOR-2 [199] datasets for the eye.

The major challenge faced in the medical imaging field is the imbalance of samples in available datasets, so there is a need to develop large-scale medical imaging datasets. The best solution is to apply multi-task learning on the deep neural network when the training data are scarce. The other possible solution is applying data augmentation techniques to the images.

### *5.5. Intelligent Transportation Systems*

The use of intelligent transportation systems (ITS) facilitates people’s lives, and they cover areas such as road sign recognition, advanced driver assistance systems, license plate recognition, vehicle detection, vehicle speed estimation, and driver awareness monitoring systems.

Both UAV and self-driving cars require real-time accurate traffic sign recognition for the safety of both passengers and surroundings. Hu et al. [200] proposed a branch output mechanism into a deep CNN to speed up traffic sign recognition. Shao et al. [201] proposed a CNN with input fed from the simplified Gabor feature map to classify the traffic signs on Chinese and German databases. Shao et al. [202] proposed "improved Faster RCNN for traffic sign detection" in real traffic situations using a "highly possible regions proposal network (HP-RPN)."

Cao et al. [203] proposed the classical LeNet-5 CNN model to improve traffic sign detection for intelligent vehicles. Zhang et al. [204] proposed an end-to-end improved YOLOv2 to achieve real-time Chinese traffic sign detection. Luo et al. [205] multi-task CNN to recognize all traffic signs classes which include both symbol-based and text-based signs. Li et al. [206] proposed a Faster RCNN and the MobileNet framework that can detect all categories of traffic signs, wherein color and shape information is used to refine localization of small traffic signs.

Recently deep learning-based methods were applied for “automatic license plate recognition” (ALPR), detecting driver’s drowsiness on highways, traffic violations, etc. Masood et al. [207] proposed license plate recognition using a sequence of deep CNNs, and the system is robust under different conditions (lighting, occlusions, variations in the pose). Laroca et al. [208] proposed an efficient YOLO detector for automatic License plate recognition, and it is robust under different conditions. Chen [209] proposed the ALPR system via sliding window + darknet YOLO tested on the AOLP dataset. Raza et al. [210] proposed a multi-channel CNN with an aggression module and SVM to achieve a high recognition rate on multi-national vehicle license plates under various illumination conditions. Goncalves et al. [211] implemented real-time ALPR via two deep multi-task networks.

An accurate perception of its surroundings is necessary for an autonomous vehicle (AV). To enable autonomous driving using deep learning techniques, one should convert collected data from sensors into semantic information. For the autonomous driving system, 3D object detection is preferred over 2D object detection, since the third dimension provides more detailed information on size and exact object locations.

Pham et al. [212] extended the “3DOP proposal generation considering class-independent proposals, then re-rank[ed] the proposals” using both monocular images and depth maps. Li et al. [213] used "a cylindrical projection mapping and a fully-convolutional network (FCN) to predict 3D bounding boxes around vehicles only" Qi et al. [214] proposed that, "Frustum Point-Net generates region proposals on the image plane with monocular images and use[s] the point cloud" to perform classification and bounding box regression.

### *5.6. Crowd Detection*

One of the most challenging tasks in object detection applications is crowd detection. In a crowded crisis, there are large crowds of confused people, resulting in pushing, mass-panic, crowd crushing, and loss of control [215]. To prevent these fatalities, automatic detection of critical and unusual situations in a dense crowd is necessary. As a result, that definitely will help to make emergency controls and appropriate decisions for security and safety [216]. This system can be used for detection and to count people, and also produce alarms in the presence of the dense crowd. Some of the applications of crowd detection are disaster management, safety control, public area management, and visual surveillance systems.

Jones [217] implemented crowd detection by using spatial-temporal information [218] and described it as a scanning window pedestrian detector. Leibe [219] implemented pedestrian detection in crowded scenes by combining local and global features in potential top-down segmentation. Lin [220] implemented crowd detection through wavelet templates and vision-based technologies. As an example, to avoid crowd related disasters and ensure public safety automatic, detection of anomalies in crowded scenes [221] should be performed in real-time. Wang et al. [113] implemented a detector trained with repulsion loss while detecting crowd occlusion scenarios.

Arandjelovic [222] used SIFT features to detect crowds and used SVM classification, which requires a proper training set. Xiang and Gong [223] implemented abnormal crowd behavior detection using label distribution learning. Still there is much research to be done in the crowd detection area. Table 7 shows a comparison of various deep learning based methods to overcome challenges raised in object detection domains.

The commonly used datasets for evaluating crowd detection performance are WIDERFACE [71], PASCAL-VOC [5], MS-COCO [77], and Open Images [86].

### *5.7. Object Detection in Sports Videos*

Deep CNNs have shown great performance in detecting objects in still images. Due to the ImageNet [70] task introduced, detecting objects from the video (VID) shifted the task of object detection into the video domain. Object detection in sports videos contains three parts: ball detection, player detection, and action recognition. The major challenge of object detection in a sports video is that it is usually desirable to track the identities of various objects between frames.

Zao et al. [224] proposed improved YOLOv3 + K-means clustering methods to achieve better performance, especially for detecting small objects, such as a sports ball or a tennis racket. Reno et al. [225] implemented a convolutional neural network-based classifier to perform ball detection in sports videos, and the system is robust to illumination changes and flickering issues. Kang et al. [226] implemented a temporal CNN for object detection on video tubelets. Pobar et al. [227] implemented the YOLOv3 object detector to detect active players in real time, tested on a custom handball video dataset.

Pobar et al. [228] proposed Mask-RCNN + Spatiotemporal interest points in detecting active players on handball custom video datasets. The same authors [229] implemented Mask-RCNN for detecting active players and tested on recorded handball practice videos. Buric et al. [230] have given an overview of various deep CNNs used for object detection in sports videos. Acuna [231] implemented a real-time multi-person detection using YOLOv2 CNN and achieved better results on the NCAA Basketball Dataset.

### *5.8. Other Domains*

Other fields include smart homes, event detection, rain detection, species detection, and visually-impaired person assistance systems. Afif et al. [232] proposed indoor object detection of specific classes for blind and visually-impaired persons using a Deep CNN RetinaNet framework. Tapu et al. [233] proposed DEEP-SEE to detect, track, and recognize in real time, objects during navigation in an outdoor environment for visually-impaired persons. Yang et al. [234] implemented the CNN model to address rain detection. Hu et al. [235] implemented a deep CNN model for shadow detection. Yang et al. [236] presented an "event detection framework to dispose of multi-domain data." Hashmi et al. [237] implemented computer vision-based assistive technology for helping visually-impaired and blind people using Deep CNN. Buzzeli et al. [238] proposed, with Faster RCNN, a vision-based system for monitoring elderly people at home.

## **6. Approaches of Deep Learning for Object Detection**

Using the DPM model, a pooling layer was introduced to handle the deformation properties of objects by Ouyang et al. [109]. Girshick et al. [16] introduced region proposal algorithm for deep learning models. The approach involves dividing an image into smaller regions with feature vectors extracted using a deep CNN model. Linear SVM performs classification on the collected feature vectors while bounding box regression is used for object localization. Similarly, generic object detection is performed by using region-lets.

Szegedy et al. [239] implemented objection detection using deep CNN [240] by replacing AlexNet [1] last layer with the regression layer. The object mask regression method was used to perform both detection and localization tasks simultaneously. DeepMultiBox [241,242] expanded the approach of Szegedy et al. [239] to detect more than one object in an image.

The major issue is how CNN learns the feature. Zeiler et al. [34] explained the conceptualization of CNN features. For visualization of features, both convolution and deconvolution processes were applied, and this technique outperforms all other techniques [1]. They established that deep learning model performance is affected by the network depth. To perform joint classification, detection and localization of tasks were carried out by applying a multi-scale sliding window approach on the Ohn-bar model [243].

Huang et al. [244] developed, "task-driven progressive part localization (TPPL) framework for fine-grained′ localization and object detection. Swarm optimization and Spatial Pyramid pooling layer are used for the detection of objects in the image region. Salient object detection [245–249] was able to achieve great performance by using deep learning techniques.

Hao et al. [250] implemented improved Faster RCNN, which significantly achieved higher accuracy, particularly while detecting small targets and occluded objects on the KITTI dataset. Leung et al. [251] implemented optimized Faster RCNN in vehicle detection under extremely dark conditions and low illumination, on the collected dataset, which has 9003 images. They were also able to detect occluded and small targets during night time with nearly no illumination. Park et al. [252] implemented a “CNN-based person detection using infrared images for night time” intrusion warning system.

Kim et al. [253] proposed shallow CNN in PVANET instead of using deep CNN for faster object detection. This method implements pipeline architecture in a few steps and based on ROI, feature extraction, regional proposal generation, and classification. Another widely used framework for real-time object detection is YOLO [40]. This method is simple: while training and testing, it scans the image only once.

The hierarchical classification method implemented using the YOLO9000 framework was proposed by Redmon and Farhadi [42]. YOLO9000 is an improved version of YOLO, and it contains 9000 object classes. YOLO9000 is not inherently used for object detection; it combines two different datasets and performs joint training on a model that is trained with both MS-COCO and ImageNet datasets.

Wang et al. [147] implemented an approach based on an adversary network. They used the "Spatial Dropout Network and Spatial Transformer Network based on adversarial network" for generating features of occlusion and deformation [20]. Finding minute differences among inter-class object classes is needed for fine-grained object detection. Chuang et al. [254] "integrated CNN with the part-based method" by introducing a co-occurrence layer. Table 11 shows comparisons to overcome challenges and difficulties arisen while using various deep learning-based object detection methods.

Currently, to speed up object detection, various acceleration techniques are adopted. Acceleration techniques are mainly classified into three types: speed up detection pipeline, detection engine, and numerical computation, as shown in Figure 15.

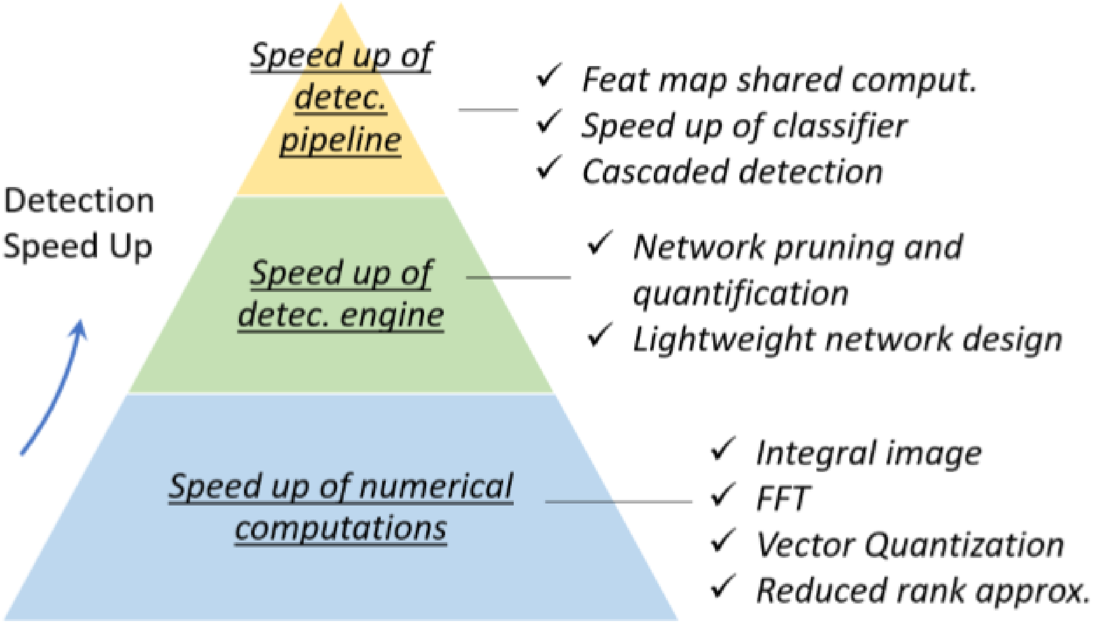
Speed up of detec. pipeline
✓ Feat map shared comput.
✓ Speed up of classifier
✓ Cascaded detection
Detection Speed Up
Speed up of detec. engine
✓ Network pruning and quantification
✓ Lightweight network design
Speed up of numerical computations
✓ Integral image
✓ FFT
✓ Vector Quantization
✓ Reduced rank approx.

**Figure 15.** Overview of acceleration techniques (source: [18]).

It is clear from this survey that deep learning-based CNNs are applicable for fine-grained object localization, detection, and generic object detection. In object detection, CNNs automatically learn the features and they form the backbone of modern object detection methods.

**Table 11.** Comparisons to overcome challenges using various deep learning-based object detection methods.

| Method | Working | Features |
|---|---|---|
| LAPGAN model (Laplacian Pyramid + Generative Adversarial Networks) [255] | Used to handle smaller images but cannot handle larger images | When training data contains smaller images with occlusions and deformations, it would generate higher resolution images. |
| Hard example mining methods [115,256] | Used to train the detectors which leads to achieve lower training loss and higher mAP value | It can use for training any object detector in order to improve detection accuracy. |
| Multi-task Deep saliency Methods [245–248] | It effectively extracts multi-scale low level and high-level features | For capturing the regions of saliency objects the network computes pixel-wise saliency values, since the pixel residing in the boundary region has similar fields so it is difficult to detect the boundaries of salient regions. Finally the network produces inaccurate map and shape of the object to be detected. |
| Adversarial Networks [147] | Uses adversarial learning and the model is invariant to deformations and occlusions by using large-scale datasets. | Since it selectively generates features and also scalable, therefore it is used for object detection in real time. |
| Example-based learning methods [92] | Used to detect objects in static images using component based model and can locate people in crowded scenes | Used to address the issue of intra-class variations in object classes, partially occluded scenes, little contrast in the background and pose variations |
| Feature-Learning Method [257–259] | Initially, each part is treated as independent object classes and works on annotated object parts during the training phase. Fine-grained categorization is the basic component during testing time. | Figure out inter-class object variations at the finer level and works more only on certain object parts. |

## 7. GPU-Based Embedded Platforms for Real Time Object Detection

The primary requirements to be fulfilled for real-time object detection using deep learning on any embedded platform are the following: higher accuracy, more speed, small model size, and better energy efficiency. The various embedded platforms available for real-time object detection using deep learning are mentioned below.

### *7.1. Raspberry Pi 4*

The latest, cheapest, and most flexible tiny product in the popular Raspberry Pi range of computers is Raspberry Pi 4 Model B. It offers great progress in processor speed, multimedia performance, memory, and connectivity compared to existing Raspberry Pi 3 Model B+. The key features of the Raspberry Pi 4 [260] module include a high-performance Broadcom BCM2711, a quad-core Cortex-A72 (ARM v8) 64-bit SoC, a pair of micro-HDMI ports which are used to connect dual displays with 4k resolution, H.265 and H.264 hardware video decoding (maximally supporting up to 4Kp60 ), 8GB RAM, dual-band 2.4/5.0 GHz IEEE 802.11ac wireless (wireless LAN), OpenGL ES, 3.0 graphics, Bluetooth 5.0, Bluetooth Low Energy (BLE), standard 40-pin GPIO, Gigabit Ethernet (2×USB 2.0 ports, 3.0 ports), a micro SD card slot for loading OS and data storage, operating temperature 40–50°, and power over Ethernet (PoE) enabled (requires separate PoE HAT).

### *7.2. ZYNQ BOARD*

For CV real-time applications in image and video processing, the Zynq-7000 family was introduced by Xilinx Company. Zynq-7000 SoC [261] is fabricated using 28nm technology and integrates both "processing system (PS) and programmable logic (PL) on a single chip." A variety of tools are available to develop code for Xilinx FPGAs, and one of the best tools for configuring FPGAs is Vivado. An FPGA family such as Virtex-7, Kintex, or Artix, uses the Vivado tool, and the same was adopted for Zynq. An advanced version of the Xilinx integrated synthesis environment (ISE) design is Vivado, which is generally used for programming FPGAs. Besides, vivado also includes a high-level synthesis (HLS) tool for C-based IP generations for high-level languages, such as "C, C++, or System C." For sequential algorithms [262], Vivado tool is used to develop optimized codes. Table 12 shows a comparison between various Zynq and Pynq boards.

**Table 12.** Comparisons between various versions of Pynq boards.

| | PYNQ-Z1 | PYNQ-Z2 | ZCU104 |
|---|---|---|---|
| Device | Zynq Z7020 | Zynq Z7020 | ZynqUltrascale+ XCZU7EV |
| Memory | 512MB DDR3 | 512MB DDR3 | 2GB DDR4, PL DDR4 SODIMM |
| Storage | $\mu$SD | $\mu$SD | $\mu$SD |
| Video | In & Out HDMI | In & Out HDMI | In & Out HDMI, Display Port |
| Audio | PDM integrated mic,<br>3.5 mm PWM audio jack | ADAU1761 codec<br>with HP + mic | - |
| Network | 10×1, ×10, ×100 Ethernet | 10×1, ×10, ×100 Ethernet | 10×1, ×10, ×100 Ethernet |
| Expansion | USB host (PS) | USB host (PS) | USB2.0/3.0 host (PS) |
| GPIO | 1× Arduino Header<br>2× Pmod<br>16× GPIO pins | 1x Arduino<br>2× Pmod<br>1× RaspberryPi | LPC FMC<br>3× Pmod (2x PL)<br>- |
| Other | 6× user LEDs<br>4× Pushbuttons<br>2× Dip switches | 6× user LEDs<br>4× Pushbuttons<br>2× Dip switches | 4× user LEDs<br>4× Pushbuttons<br>4× Dip switches |

### *7.3. NVIDIA JETSON TX2 BOARD*

Jetson TX2 [263] is one of the fastest and most power efficient CV applications. It provides an easy way to deploy hardware and software for real-time object detection. It supports NVIDIA Jetpack on a software development kit (SDK) which includes a board support package (BSP), deep learning libraries, CV applications, GPU computational power, and image and video processing.

Jetson TX2 features include high performance two Denver 64-bit CPU's + QUAD core A57, and integrated 256-core NVIDIA Pascal GPU, 8GB 128 bit DDR4 internal memory, 32 GB external memory card, 4kp60 H.264/5 encoder, and a decoder; it supports 10 ×1/×10/×100 base-T ethernet and Gigabit ethernet. It provides USB 3.0, USB 2.0, and micro USB, supports HDMI, M.2 Key E, SD, GPIOs, I2C, I2S, SPI, and Dual CAN bus, and provides on-chip TTL UART.

Jetson TX2 is useful for deploying in the CV and deep learning application areas, since it runs with open-source Linux OS and performs calculations over one teraflop. A super-computer that consumes less than 7.5 Watts on an off mode module brings a true real-time embedded AI computing device. In terms of performance and efficiency in CV applications, it has surpassed the world's most autonomous machines.

### *7.4. GPU-Based CNN Object Detection*

The GPU parallel processing capability decreases the needed processing time, allowing a better system performance when compared with the obtained CPU times. The GPU allows a programmable and parallel implementation, but we need to ensure correct synchronization and memory access. To create high-performance GPU-accelerated applications with parallel programming, a variety of development platforms, such as compute unified device architecture (CUDA) [264] and open computing language (OpenCL) [265] are utilized for GPU-accelerated embedded systems.

For a system with a single machine and multi-GPUs working on separate tasks, one can directly access any available GPU without coding in CUDA. On the other hand, for multi-GPUs working on shared tasks, such as training several models with different hyperparameters, distributed training is needed. Nvidia provides distributed training and is now supported by many popular deep learning frameworks, such as Pytorch, Caffe, TensorFlow, etc. These techniques reduce computational time linearly with the number of GPUs [266]. Nvidia Jetson is a leading low-power embedded platform that enables server-grade computing performance on edge devices.

Altera's Cyclone V FPGA achieves a speed of 24.3 fps, for real-time lane detection using Hough transform algorithm [267]. Object detection and tracking system using the FPGA Zynq XC7Z020 board achieves a speed of 30 fps using a modified background subtraction algorithm [268]. Real-time object detection on Zynq UltraScale + MPSoC zcu102 evaluation board using a fully pipelined binarized deep convolutional neural network (BCNN) achieved a processing speed of 34.9 fps [261]. YOLOv2 object detection algorithm running on an NVIDIA Jetson TX2 achieved a processing speed of 9.57 fps [269].

Hossain et al. [270] implemented real-time object detection and tracking from a UAV and tested it on a Jetson Xavier board using various deep learning techniques and has it compared with Jetson variants. Stepanenko et al. [271] implemented YOLO with TensorRT 5.0; it reduces network size which in turn increases speed compared with thw YOLO + Darknet model implementation. Korez et al. [272] proposed a combination of Faster RCNN + deformable convolution + FPN + weight standardization techniques for object detection on low capacity GPU systems. Cambay et al. [273] implemented a Deep CNN YOLO object detector on both the PYNQ FPGA board and USB-GPU called Movidius GPU-based accelerated system.

Table 13 shows the performance results in terms of speed; i.e., seconds per frame (FPS). A quantitative comparison between Jetson variants and GPU-based offline work station shows that the comparison changes with the dimensions of input image. One can choose the best algorithm and system for a specific application with the help of this table.

**Table 13.** Performance comparison between Jetson modules and GTX 1080 for real-time object detection.

| S.No | Architecture | TX1 (Fps) | TX2 (Fps) | Xavier AGX (Fps) | GPU-Based Work Station Gtx 1080 (Fps) |
|---|---|---|---|---|---|
| 1 | YOLOv2 | 3 | 10 | 25–30 | 27 |
| 2 | YOLOv3 | — | 4 | 15–18 | 15.8 |
| 3 | Tiny YOLOv3 | 8–10 | 11 | 31 | 31+ |
| 4 | SSD | 8 | 10–12 | 34–49 | 33 |
| 5 | Faster RCNN | — | 1 | 1.2 | — |
| 6 | Mask RCNN | — | — | — | 3–4 |

## 8. Research Directions

Despite great progress achieved in the object detection field, still, the technology remains significantly far away from human vision while addressing real-world challenges, such as: detecting objects under constrained conditions, working in an open world, and other modalities.

We can see the following directions of future research, based on these challenges:

1. More efficient detection frameworks: The main reason for the success of object detection is due to the development of superior detection frameworks, both in two-stage and one-stage detectors (RCNN, Fast/Faster/Mask RCNN, YOLO, and SSD). Two-stage detectors exhibit high accuracy, whereas single-stage detectors are simple and faster. Object detectors depend a lot on the underlying backbone models, and most of them are optimized for classification of images, possibly causing a learning bias; and it could be helpful to develop new object detectors learning from scratch.
2. Compact and efficient CNN features: CNN layers are increased in depth from several layers (AlexNet) to hundreds of layers (ResNet, ResNext, CentreNet, DenseNet). All these networks require a lot of data and high-end GPUs for training, since they have billions of parameters. Thus, to reduce network redundancy further, researchers should show interest in designing lightweight and compact networks.
3. Weakly supervised detection: At present all the state-of-the-art detectors use only labeled data with either object segmentation masks or bounding boxes on the fully supervised models. But in the absence of labeled training data, fully supervised learning is not scalable, so it is essential to design a model where only partially labeled data are available.
4. Efficient backbone architecture for object detection: Done by adopting weights of pre-trained classification models, since they are trained on large-scale datasets for object detection tasks. Thus, adopting a pre-trained model might not result in an optimal solution due the conflicts between image classification and object detection tasks. Currently, most object detectors are based on classification backbones, and only a few use different backbone models (like SqueezeDet based on SqueezeNet). So there is a need to develop a detection-aware light-weight backbone model for real-time object detection.
5. Object detection in other modalities: Currently most of the object detectors work only with 2D images, but detection in other modalities—3D, LIDAR, etc.—would be highly relevant in application areas such as self-driving cars [274], drones, and robots. However, again the 3D object detection may raise new challenges using video, depth, and cloud points.
6. Network optimization: Selecting an optimal detection network brings a perfect balance between speed, memory, and accuracy for a specific application and on embedded hardware. Though the detection accuracy is reduced, it is better to teach compact models with few parameters, and this situation might be overcome by introducing hint learning, knowledge distillation, and better pre-training schemes.
7. Scale adaption [19]: It is more obvious in face detection and crowd detection; objects usually exist on different scales. In order to increase the robustness to learn spatial transformation, it is necessary to train designed detectors in scale-invariant, multi-scale, or scale-adaptive ways.

(a) For scale-adaptive detectors, make an attention mechanism, form a cascaded network, and scale a distribution estimation for detecting objects adaptively.
(b) For multi-scale detectors, both the GAN (generative adversarial network) and FPN (feature pyramid network) generate a multi-scale feature map.
(c) For scale-invariant detectors, reverse connection, hard negative mining, backbone models such as AlexNet (rotation invariance) and ResNet are all beneficial.

8. Cross-dataset training [275]: Cross-dataset training for object detection aims to detect the union of all the classes across different existing datasets with a single model and without additional labeling, which in turn saves the heavy burden of labeling new classes on all the existing datasets. Using cross-dataset training, one only needs to label the new classes on the new dataset. It is widely used in industrial applications that are usually faced with increasing classes.

Object detection is independent of domains and the research in many fields is still far complete.

## 9. Conclusions and Future Scope

Applications like machine vision and self-driving cars [276] consider object detection as the fundamental step. This paper comprehensively reviews both traditional object detectors and deep learning-based object detectors starting from RCNN and going all the way to the latest CornerNet, with its pros and cons. We have summarized different object detectors' performances on MS-COCO and Pascal-VOC datasets. Various deep learning frameworks and available API services for object detection are briefly summarized. Since all CV algorithms are trained and verified using benchmark datasets only, we covered widely used datasets for specific applications; datasets play a crucial role in achieving the correct output.

Image classification dataset challenges' statistics, and also PASCAL-VOC and ILSRVC comparisons, which are released by worldwide competitions, are also presented. Deep learning-based object detection technology is developing rapidly day to day due to upgrading computational power. In order to deploy deep learning-based object detectors in applications like self-driving cars, high-precision, real-time systems are urgently needed.

The ultimate goal while deploying deep learning-based object detectors to real-time scenarios is to achieve both higher accuracy and speed. Researchers are moving in this direction and have improved processing speed, constructed new architectures and lightweight feature extractors, solved complex environment scenes like small and occluded objects, improved localization accuracy, improved confidence scores, enhanced post-processing methods, made anchor-less detectors to overcome the data imbalance problem during training, done object detection with the latest engines, detected with better features, i.e., Feature Fusion, designed new loss functions, enhanced object detection by semantic segmentation, performed scale-adaptive detection, trained from scratch, and achieved better results by combining both one-stage and two-stage detectors.

With the day to day increase of powerful, deep learning-based object detectors, we covered their use in fields such as pedestrian detection, face detection, the military, medical image analysis, intelligent transporation systems, crowd detection, and sports, and still, applications in the object detection domain are arising. Although there is great success in this domain, there is much scope for further development. Comparisons to overcome challenges and difficulties raised using various deep learning-based object detection methods and speed up techniques are also covered. In this paper, various GPU-based embedded platforms and a quantitative performance comparison between Jetson variants and GPU-based offline work station for real-time object detection using deep learning techniques are also assessed.

Real-time object detection using different GPU-based embedded platforms should be robust with respect to invariant occlusions, scales, illumination, intra-class variations, and deformations. Due to these factors, there are high chances of not detecting small objects in a video, which reduces the performance of real-time object detection systems.

Object detection is a fundamental problem to be solved; existing methods were developed. But still, there is a huge scope for developing new mechanisms and object detection as basic services in real-time applications, such as deep-sea bases, driverless cars, robots navigating on planets, industrial plants, and drone cameras where high precision is expected for certain tasks.

Particularly, for detecting some small objects there remains a large speed gap between human eyes and machine vision. The future of object detection would be AutoML; i.e., designing a detection model to reduce human intervention. For more accurate detection, much data is required. To improve overall accuracy further, training images with more diversity (scale, view-angle) of the object are needed. Finally, we point out that promising future directions in this research field are not limited to the aforementioned aspects, and the research in this field is still far from complete.

**Author Contributions:** Conceptualization, C.B.M. and M.F.H.; methodology, C.B.M. and M.F.H.; software, C.B.M., M.F.H., and N.D.B.; validation, C.B.M., M.F.H., and N.D.B.; formal analysis, C.B.M., M.F.H., and N.D.B.; investigation, C.B.M. and M.F.H.; resources, C.B.M. and M.F.H.; data curation, C.B.M. and M.F.H.; writing—original draft preparation, C.B.M., M.F.H., and N.D.B.; writing—review and editing, C.B.M., M.F.H., N.D.B., and Z.W.G.; visualization, M.F.H. and N.D.B.; supervision, N.D.B., M.F.H., and Z.W.G.; project administration, C.B.M., M.F.H., N.D.B., and Z.W.G.; funding acquisition, Z.W.G. and N.D.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Energy Cloud R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT [2019M3F2A1073164].

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| 2D | 2-dimensional |
| ALPR | Automatic License Plate Recognition |
| AP | Average Precision |
| API | Application Program Interface |
| ARD | Acute Respiratory Disease |
| ARM | Advanced RISC Machine |
| AV | Autonomous Vehicle |
| BB | Bounding Box |
| BCNN | Binarized deep Convolutional Neural Network |
| BLE | Bluetooth Low Energy |
| BSP | Board Support Package |
| CAD | Computer Aided Design |
| CAN | Controller Area Network |
| CNN | Convolutional Neural Network |
| COCO | Common Objects in Context |
| COPD | Chronic Obstructive Pulmonary Disease |
| CPU | Central Processing Unit |
| CUDA | Compute Unified Device Architecture |
| CV | Computer Vision |
| DBN | Deep Belief Network |
| DCNN | Deep Convolutional Neural Network |
| DDR | Double Data Rate |
| DNN | Deep Neural Network |
| DPM | Deformable Part-based Model |
| FC | Fully-Convolutional |
| FCN | Fully-Convolutional Network |
| FDDB | Face Detection Dataset and Benchmark |
| FPGA | Field Programmable Gate Array |
| FPN | Feature Pyramid Networks |
| FPS | Frames Per Second |

| | |
|---|---|
| GAN | Generative Adversarial Network |
| GPIO | General Purpose Input/Output |
| GPU | Graphics Processing Unit |
| HCI | Human-Computer Interaction |
| HDMI | High-Definition Multimedia Interface |
| HLS | HTTP Live Streaming |
| HOG | Histogram of Oriented Gradients |
| HTTP | Hypertext Transfer Protocol |
| I2C | Inter-IC |
| I2S | Inter-IC Sound |
| ICF | Integral Channel Features |
| IEEE | Institute of Electrical and Electronics Engineers |
| ILSVRC | ImageNet Large Scale Visual Recognition Challenge |
| IoT | Internet-of-things |
| IOU | Intersection over Union |
| ISE | Integrated Synthesis Environment |
| ITS | Intelligent Transportation Systems |
| LAN | Local Area Network |
| mAP | mean Average Precision |
| MRI | Magnetic Resonance Imaging |
| MS-COCO | Microsoft Common Objects in Context |
| PASCAL | a Procedural Programming Language |
| PCI | Peripheral Component Interconnect |
| PL | Programmable Logic |
| PoE | Power over Ethernet |
| PS | Processing System |
| PVANET | a lightweight feature extraction network architecture for object detection |
| RAM | Random Access Memory |
| RBM | Restricted Boltzmann Machine |
| RCNN | Regions with Convolutional Neural Netwoks |
| REST | Representational State Transfer |
| RFCN | Region-based Fully-Convolutional Networks |
| RGB | Red Green Blue |
| ROI | Region of Interest |
| RPN | Region Proposal Network |
| SVM | Support Vector Machine |
| SD | Secure Digital |
| SDK | Software Development Kit |
| SIFT | Scale-Invariant Feature Transform |
| SoC | System-on-Chip |
| SPI | Serial Peripheral Interface |
| SPP | Spatial Pyramid Pooling |
| SPPNet | Spatial Pyramid Pooling Network |
| SSD | Single Shot Multi-Box Detector |
| SUN | Scene UNderstanding dataset |
| SVM | Support Vector Machine |
| TPPL | Task-driven Progressive Part Localization |
| TTL | Transistor-Transistor Logic |
| UAV | Unmanned Aerial Vehicles |
| USB | Universal Serial Bus |
| VHR | Very High Resolution |
| VJ | Viola–Jones |
| VOC | Visual Object Classes Challenge |
| YOLO | You Only Look Once |

## References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems; Curran Associates, Inc., San Diego, CA, 2012*; pp. 1097–1105.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: San Diego, CA, USA, 2015; pp. 91–99.
3. Nguyen, H.T.; Lee, E.H.; Lee, S. Study on the Classification Performance of Underwater Sonar Image Classification Based on Convolutional Neural Networks for Detecting a Submerged Human Body. *Sensors* **2020**, *20*, 94.
4. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252.
5. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338.
6. Fourie, J.; Mills, S.; Green, R. Harmony filter: A robust visual tracking system using the improved harmony search algorithm. *Image Vis. Comput.* **2010**, *28*, 1702–1716.
7. Cuevas, E.; Ortega-Sánchez, N.; Zaldivar, D.; Pérez-Cisneros, M. Circle detection by harmony search optimization. *J. Intell. Robot. Syst.* **2012**, *66*, 359–376.
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
9. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507.
10. McIvor, A.M. Background subtraction techniques. *Proc. Image Vis. Comput.* **2000**, *4*, 3099–3104.
11. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. I.
12. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154.
13. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Sonference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
14. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
15. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D. Cascade object detection with deformable part models. In Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 13–18 June 2010; pp. 2241–2248.
16. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645.
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, 24–27 June 2014; pp. 580–587.
18. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055.
19. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318.
20. Pathak, A.R.; Pandey, M.; Rautaray, S. Application of deep learning for object detection. *Procedia Comput. Sci.* **2018**, *132*, 1706–1717.
21. Sultana, F.; Sufian, A.; Dutta, P. A review of object detection models based on convolutional neural network. *arXiv* **2019**, arXiv:1905.01614.
22. Zhao, Z.Q.; Zheng, P.; Xu, S.t.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232.
23. Mittal, U.; Srivastava, S.; Chawla, P. Review of different techniques for object detection using deep learning. In Proceedings of the Third International Conference on Advanced Informatics for Computing Research, Shimla, India, 15–16 June 2019; pp. 1–8.

24. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, 20–25 September 1999; Volume 2, pp. 1150–1157.
25. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
26. Belongie, S.; Malik, J.; Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522.
27. Girshick, R.B.; Felzenszwalb, P.F.; Mcallester, D.A. Object detection with grammar models. In *Advances in Neural Information Processing System*s; Curran Associates Inc.: San Francisco, CA, USA, 2011; pp. 442–450.
28. Girshick, R.B. From Rigid Templates to Grammars: Object Detection with Structured Models. Ph.D. Thesis, The University of Chicago, Chicago, IL, USA, 2012.
29. Li, Y.F.; Kwok, J.T.; Tsang, I.W.; Zhou, Z.H. A convex method for locating regions of interest with multi-instance learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Antwerp, Belgium 14–18 September 2009; Springer: Berlin, Germany, 2009; pp. 15–30.
30. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171.
31. Girshick, R.B.; Felzenszwalb, P.F.; McAllester, D. Discriminatively Trained Deformable Part Models, Release 5. 2012. Available online: http://people.cs.uchicago.edu/~rbg/latent-release5/ (accessed on 7 May 2020).
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
33. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.
34. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Nancy, France, 14–18 September 2014; Springer: Berlin, Germany, 2014; pp. 818–833.
35. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: San Francisco, CA, USA, 2016; pp. 379–387.
36. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-head R-CNN: In defense of two-stage object detector. *arXiv* **2017**, arXiv:1711.07264.
37. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
38. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 26 June–1 July 2016; pp. 770–778.
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
41. 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 European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 Octomber 2016; Springer: Berlin, Germany, 2016; pp. 21–37.
42. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
43. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
44. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the International Cconference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
45. Wu, B.; Iandola, F.; Jin, P.H.; Keutzer, K. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 129–137.
46. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.

47. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, UT, USA, 18–23 June 2018; pp. 3974–3983.
48. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
49. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Object detection with keypoint triplets. *arXiv* **2019**, arXiv:1904.08189.
50. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
51. Mathematica. Available online: https://www.wolfram.com/mathematica/ (accessed on 31 December 2019).
52. Dlib. Available online: Dlib.net (accessed on 31 December 2019).
53. Theano. Available online: http://deeplearning.net/software/theano/ (accessed on 31 December 2019).
54. Caffe. Available online: http://caffe.berkeleyvision.org/ (accessed on 31 December 2019).
55. Deeplearning4j. Available online: https://deeplearning4j.org (accessed on 31 December 2019).
56. Cahiner. Available online: https://chainer.org (accessed on 31 December 2019).
57. Keras. Available online: https://keras.io/ (accessed on 31 December 2019).
58. Mathworks—Deep Learning. Available online: https://in.mathworks.com/solutions/deep-learning.html (accessed on 31 December 2019).
59. Apache. Available online: http://singa.apache.org (accessed on 31 December 2019).
60. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 31 December 2019).
61. Pytorch. Available online: https://pytorch.org (accessed on 31 December 2019).
62. BigDL. Available online: https://github.com/intel-analytics/BigDL (accessed on 31 December 2019).
63. Apache. Available online: http://www.apache.org (accessed on 31 December 2019).
64. MXnet. Available online: http://mxnet.io/ (accessed on 31 December 2019).
65. Microsoft Cognitive Service. Available online: https://www.microsoft.com/cognitive-services/en-us/computer-vision-api (accessed on 31 December 2019).
66. Amazon Recognition. Available online: https://aws.amazon.com/rekognition/ (accessed on 31 December 2019).
67. IBM Watson Vision Recognition service. Available online: http://www.ibm.com/watson/developercloud/visual-recognition.html (accessed on 31 December 2019).
68. Google Cloud Vision API. Available online: https://cloud.google.com/vision/ (accessed on 31 December 2019).
69. Cloud Sight. Available online: https://cloudsight.readme.io/v1.0/docs (accessed on 31 December 2019).
70. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
71. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. Wider face: A face detection benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, Nevada, USA, 26 June–1 July 2016; pp. 5525–5533.
72. Jain, V.; Learned-Miller, E. *Fddb: A Benchmark for Face Detection in Unconstrained Settings*; Technical Report, UMass Amherst Technical Report; UMass Amherst Libraries: Amherst, MA, USA, 2010.
73. Zhang, S.; Benenson, R.; Schiele, B. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21–26 July 2017; pp. 3213–3221.
74. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 3213–3223.
75. Ess, A.; Leibe, B.; Van Gool, L. Depth and appearance for mobile scene analysis. In Proceedings of the 2007 IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
76. Torralba, A.; Fergus, R.; Freeman, W.T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1958–1970.
77. 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 European Conference on Computer Vision, Zurich, Switzerland, 5–12 September 2014; Springer: Berlin, Germany, 2014; pp. 740–755.

78. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, University of Tront, Toronto, ON, Canada, 2009.
79. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S.; Goering, C.; Berg, T.; Belhumeur, P. *Caltech-UCSD Birds-200-2011*; California Institute of Technology: Pasadena, CA, USA, 2011.
80. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD birds 200*; California Institute of Technology: Pasadena, CA, USA, 2010.
81. Griffin, G.; Holub, A.; Perona, P. *Caltech-256 Object Category Dataset*; California Institute of Technology: Pasadena, CA, USA, 2007. '
82. ILSVRC Detection Challenge Results. Available online: http://www.image-net.org/challenges/LSVRC/ (accessed on 31 December 2019).
83. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136.
84. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173.
85. Xiao, J.; Hays, J.; Ehinger, K.A.; Oliva, A.; Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3485–3492.
86. Open Images. Available online: https://www.kaggle.com/bigquery/open-images (accessed on 31 December 2019).
87. Kragh, M.F.; Christiansen, P.; Laursen, M.S.; Larsen, M.; Steen, K.A.; Green, O.; Karstoft, H.; Jørgensen, R.N. FieldSAFE: Dataset for obstacle detection in agriculture. *Sensors* **2017**, *17*, 2579.
88. Grady, N.W.; Underwood, M.; Roy, A.; Chang, W.L. Big data: Challenges, practices and technologies: NIST big data public working group workshop at IEEE big data 2014. In Proceedings of the International Conference on Big Data, Washington DC, USA, 27–30 October 2014; pp. 11–15.
89. Dollár, P.; Tu, Z.; Perona, P.; Belongie, S. *Integral Channel Features*; BMVC Press: London, UK, 2009.
90. Maji, S.; Berg, A.C.; Malik, J. Classification using intersection kernel support vector machines is efficient. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
91. Zhu, Q.; Yeh, M.C.; Cheng, K.T.; Avidan, S. Fast human detection using a cascade of histograms of oriented gradients. In Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1491–1498.
92. Mohan, A.; Papageorgiou, C.; Poggio, T. Example-based object detection in images by components. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 349–361.
93. Wang, X.; Han, T.X.; Yan, S. An HOG-LBP human detector with partial occlusion handling. In Proceedings of the International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 32–39.
94. Wu, B.; Nevatia, R. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In Proceedings of the International Conference on Computer Vision, Beijing, China, 17–21 October 2005; Volume 1, pp. 90–97.
95. Andreopoulos, A.; Tsotsos, J.K. 50 years of object recognition: Directions forward. *Comput. Vis. Image Underst.* **2013**, *117*, 827–891.
96. Sadeghi, M.A.; Forsyth, D. 30hz object detection with dpm v5. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014; pp. 65–79.
97. Hosang, J.; Omran, M.; Benenson, R.; Schiele, B. Taking a deeper look at pedestrians. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4073–4082.
98. Yi, Z.; Yongliang, S.; Jun, Z. An improved tiny-yolov3 pedestrian detection algorithm. *Optik* **2019**, *183*, 17–23.
99. Zhang, L.; Lin, L.; Liang, X.; He, K. Is faster R-CNNN doing well for pedestrian detection? In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 443–457.
100. Song, T.; Sun, L.; Xie, D.; Sun, H.; Pu, S. Small-scale pedestrian detection based on somatic topology localization and temporal feature aggregation. *arXiv* **2018**, arXiv:1807.01438.
101. Cao, J.; Pang, Y.; Li, X. Learning multilayer channel features for pedestrian detection. *IEEE Trans. Image Process.* **2017**, *26*, 3210–3220.

102. Mao, J.; Xiao, T.; Jiang, Y.; Cao, Z. What can help pedestrian detection? In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3127–3136.
103. Krishna, H.; Jawahar, C. Improving small object detection. In Proceedings of the 4th IAPR Asian Conference on Pattern Recognition, Nanjing China, 26–29 November ACPR, 2017; pp. 340–345.
104. Hu, Q.; Wang, P.; Shen, C.; van den Hengel, A.; Porikli, F. Pushing the limits of deep cnns for pedestrian detection. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 1358–1368.
105. Lee, Y.; Bui, T.D.; Shin, J. Pedestrian detection based on deep fusion network using feature correlation. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 12–15 November 2018; pp. 694–699.
106. Cai, Z.; Saberian, M.; Vasconcelos, N. Learning complexity-aware cascades for deep pedestrian detection. In Proceedings of the IEEE International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 3361–3369.
107. Bosquet, B.; Mucientes, M.; Brea, V.M. STDnet: Exploiting high resolution feature maps for small object detection. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103615.
108. Tian, Y.; Luo, P.; Wang, X.; Tang, X. Deep learning strong parts for pedestrian detection. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1904–1912.
109. Ouyang, W.; Zhou, H.; Li, H.; Li, Q.; Yan, J.; Wang, X. Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1874–1887.
110. Zhang, S.; Yang, J.; Schiele, B. Occluded pedestrian detection through guided attention in CNNs. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6995–7003.
111. Gao, M.; Yu, R.; Li, A.; Morariu, V.I.; Davis, L.S. Dynamic zoom-in network for fast object detection in large images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6926–6935.
112. Lu, Y.; Javidi, T.; Lazebnik, S. Adaptive object detection using adjacency and zoom prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2351–2359.
113. Wang, X.; Xiao, T.; Jiang, Y.; Shao, S.; Sun, J.; Shen, C. Repulsion loss: Detecting pedestrians in a crowd. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7774–7783.
114. Tian, Y.; Luo, P.; Wang, X.; Tang, X. Pedestrian detection aided by deep learning semantic tasks. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5079–5087.
115. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 761–769.
116. Tang, T.; Zhou, S.; Deng, Z.; Zou, H.; Lei, L. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors* **2017**, *17*, 336.
117. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
118. Jin, J.; Fu, K.; Zhang, C. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1991–2000.
119. Zhou, M.; Jing, M.; Liu, D.; Xia, Z.; Zou, Z.; Shi, Z. Multi-resolution networks for ship detection in infrared remote sensing images. *Infrared Phys. Technol.* **2018**, *92*, 183–189.
120. Xu, D.; Ouyang, W.; Ricci, E.; Wang, X.; Sebe, N. Learning cross-modal deep representations for robust pedestrian detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5363–5371.
121. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Occlusion-aware R-CNN: Detecting pedestrians in a crowd. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 637–653.

122. Zhou, C.; Yuan, J. Bi-box regression for pedestrian detection and occlusion estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 135–151.
123. Hsu, W.Y. Automatic pedestrian detection in partially occluded single image. *Integr. Comput.-Aided Eng.* **2018**, *25*, 369–379.
124. Ren, Y.; Zhu, C.; Xiao, S. Deformable faster r-cnn with aggregating multi-layer features for partially occluded object detection in optical remote sensing images. *Remote Sens.* **2018**, *10*, 1470.
125. Li, W.; Ni, H.; Wang, Y.; Fu, B.; Liu, P.; Wang, S. Detection of partially occluded pedestrians by an enhanced cascade detector. *IET Intell. Transp. Syst.* **2014**, *8*, 621–630.
126. Yang, G.; Huang, T.S. Human face detection in a complex background. *Pattern Recognit.* **1994**, *27*, 53–63.
127. Craw, I.; Tock, D.; Bennett, A. Finding face features. In Proceedings of the European Conference on Computer Vision, Santa Margherita Ligure, Italy, 9–22 May 1992; Springer: Berlin, Germany, 1992; pp. 92–96.
128. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86.
129. Vaillant, R.; Monrocq, C.; Le Cun, Y. Original approach for the localisation of objects in images. *IEE Proc. Vision, Image Signal Process.* **1994**, *141*, 245–250.
130. Pentland.; Moghaddam.; Starner. View-based and modular eigenspaces for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, DC, USA, 21–23 June 1994; pp. 84–91.
131. Rowley, H.A.; Baluja, S.; Kanade, T. Human face detection in visual scenes. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: San Francisco, CA, USA, 1996; pp. 875–881.
132. Rowley, H.A.; Baluja, S.; Kanade, T. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 23–38.
133. Osuna, E.; Freund, R.; Girosit, F. Training support vector machines: An application to face detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, 17–19 June 1997; pp. 130–136.
134. Byun, H.; Lee, S.W. Applications of support vector machines for pattern recognition: A survey. In Proceedings of the International Workshop on Support Vector Machine, Niagara Falls, ON, Canada, 10 August 2002; Springer: Berlin, Germany, 2002; pp. 213–236.
135. Xiao, R.; Zhu, L.; Zhang, H.J. Boosting chain learning for object detection. In Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 14–17 October 2003; pp. 709–715.
136. Zhang, Y.; Zhao, D.; Sun, J.; Zou, G.; Li, W. Adaptive convolutional neural network and its application in face recognition. *Neural Process. Lett.* **2016**, *43*, 389–399.
137. Wu, S.; Kan, M.; Shan, S.; Chen, X. Hierarchical Attention for Part-Aware Face Detection. *Int. J. Comput. Vis.* **2019**, *127*, 560–578.
138. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, MA, USA, 7–12 June 2015; pp. 5325–5334.
139. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503.
140. Hao, Z.; Liu, Y.; Qin, H.; Yan, J.; Li, X.; Hu, X. Scale-aware face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6186–6195.
141. Najibi, M.; Samangouei, P.; Chellappa, R.; Davis, L.S. SSH: Single stage headless face detector. In Proceedings of the IEEE International Conference on Computer Vision,Honolulu, HI, USA, 21–26 July 2017; pp. 4875–4884.
142. Shi, X.; Shan, S.; Kan, M.; Wu, S.; Chen, X. Real-time rotation-invariant face detection with progressive calibration networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 July 2018; pp. 2295–2303.
143. Chen, D.; Hua, G.; Wen, F.; Sun, J. Supervised transformer network for efficient face detection. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 122–138.
144. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. Faceness-net: Face detection through deep facial part responses. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1845–1859.
145. Ghodrati, A.; Diba, A.; Pedersoli, M.; Tuytelaars, T.; Van Gool, L. Deepproposal: Hunting objects by cascading deep convolutional layers. In Proceedings of the IEEE International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015 ; pp. 2578–2586.

146. Wang, J.; Yuan, Y.; Yu, G. Face attention network: An effective face detector for the occluded faces. *arXiv* **2017**, arXiv:1711.07246.
147. Wang, X.; Shrivastava, A.; Gupta, A. A-fast-RCNN: Hard positive generation via adversary for object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2606–2615.
148. Zhou, Y.; Liu, D.; Huang, T. Survey of face detection on low-quality images. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition, Xi'an, China, 15–19 May 2018; pp. 769–773.
149. Yang, S.; Xiong, Y.; Loy, C.C.; Tang, X. Face detection through scale-friendly deep convolutional networks. *arXiv* **2017**, arXiv:1706.02863.
150. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. S3fd: Single shot scale-invariant face detector. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 192–201.
151. Cai, Z.; Fan, Q.; Feris, R.S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. In Proceedings of the European conference on computer vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 354–370.
152. Zhang, C.; Xu, X.; Tu, D. Face detection using improved faster rcnn. *arXiv* **2018**, arXiv:1802.02142.
153. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z. Scale-aware trident networks for object detection. In Proceedings of the IEEE International Conference on Computer Vision, South Korea, 27 October–2 November 2019; pp. 6054–6063.
154. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Detnet: A backbone network for object detection. *arXiv* **2018**, arXiv:1804.06215.
155. Liu, S.; Huang, D.; Wang, Y. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
156. Zhang, F.; Du, B.; Zhang, L.; Xu, M. Weakly supervised learning based on coupled convolutional neural networks for aircraft detection. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5553–5563.
157. Han, J.; Zhang, D.; Cheng, G.; Guo, L.; Ren, J. Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Trans. Geosci. Remote. Sens.* **2014**, *53*, 3325–3337.
158. Li, Q.; Wang, Y.; Liu, Q.; Wang, W. Hough transform guided deep feature extraction for dense building detection in remote sensing images. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 1872–1876.
159. Mou, L.; Zhu, X.X. Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6699–6711.
160. Chen, X.; Xiang, S.; Liu, C.L.; Pan, C.H. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1797–1801.
161. Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N.; Zuair, M. Deep learning approach for car detection in UAV imagery. *Remote Sens.* **2017**, *9*, 312.
162. Ma, W.; Guo, Q.; Wu, Y.; Zhao, W.; Zhang, X.; Jiao, L. A novel multi-model decision fusion network for object detection in remote sensing images. *Remote Sens.* **2019**, *11*, 737.
163. Zhang, X.; Zhu, K.; Chen, G.; Tan, X.; Zhang, L.; Dai, F.; Liao, P.; Gong, Y. Geospatial object detection on high resolution remote sensing imagery based on double multi-scale feature pyramid network. *Remote Sens.* **2019**, *11*, 755.
164. Wang, J.; Ding, J.; Guo, H.; Cheng, W.; Pan, T.; Yang, W. Mask OBB: A Semantic Attention-Based Mask Oriented Bounding Box Representation for Multi-Category Object Detection in Aerial Images. *Remote Sens.* **2019**, *11*, 2930.
165. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote. Sens.* **2016**, *54*, 7405–7415.
166. Li, Q.; Mou, L.; Xu, Q.; Zhang, Y.; Zhu, X.X. R3-net: A deep network for multi-oriented vehicle detection in aerial images and videos. *arXiv* **2018**, arXiv:1808.05560.
167. Pang, J.; Li, C.; Shi, J.; Xu, Z.; Feng, H. R2-CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 5512–5524.

168. Qian, X.; Lin, S.; Cheng, G.; Yao, X.; Ren, H.; Wang, W. Object Detection in Remote Sensing Images Based on Improved Bounding Box Regression and Multi-Level Features Fusion. *Remote Sens.* **2020**, *12*, 143.
169. Cheng, G.; Han, J.; Zhou, P.; Guo, L. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote. Sens.* **2014**, *98*, 119–132.
170. Liu, K.; Mattyus, G. Fast multiclass vehicle detection on aerial images. *IEEE Geosci. Remote. Sens. Lett.* **2015**, *12*, 1938–1942.
171. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203.
172. Zhang, Y.; Yuan, Y.; Feng, Y.; Lu, X. Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 5535–5548.
173. Islam, J.; Zhang, Y. Early Diagnosis of Alzheimer's Disease: A Neuroimaging Study with Deep Learning Architectures. In Proceedings of the IEEE Conference on Computer vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 19–21 June 2018; pp. 1881–1883.
174. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-first AAAI conference on artificial intelligence, San Francisco, CA, USA, 4–9 February 2017.
175. Marcus, D.S.; Fotenos, A.F.; Csernansky, J.G.; Morris, J.C.; Buckner, R.L. Open access series of imaging studies: Longitudinal MRI data in nondemented and demented older adults. *J. Cogn. Neurosci.* **2010**, *22*, 2677–2684.
176. Alaverdyan, Z.; Jung, J.; Bouet, R.; Lartizien, C. Regularized siamese neural network for unsupervised outlier detection on brain multiparametric magnetic resonance imaging: Application to epilepsy lesion screening. *Med Image Anal.* **2020**, *60*, 101618.
177. Laukamp, K.R.; Thiele, F.; Shakirin, G.; Zopfs, D.; Faymonville, A.; Timmer, M.; Maintz, D.; Perkuhn, M.; Borggrefe, J. Fully automated detection and segmentation of meningiomas using deep learning on routine multiparametric MRI. *Eur. Radiol.* **2019**, *29*, 124–132.
178. Katzmann, A.; Muehlberg, A.; Suehling, M.; Noerenberg, D.; Holch, J.W.; Heinemann, V.; Gross, H.M. Predicting Lesion Growth and Patient Survival in Colorectal Cancer Patients Using Deep Neural Networks. In Proceedings of the Conference track: Medical Imaging with Deep Learning, Amsterdam, The Netherlands, 4–6 July 2018.
179. Bejnordi, B.E.; Veta, M.; Van Diest, P.J.; Van Ginneken, B.; Karssemeijer, N.; Litjens, G.; Van Der Laak, J.A.; Hermsen, M.; Manson, Q.F.; Balkenhol, M.; et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA* **2017**, *318*, 2199–2210.
180. Zhang, J.; Cain, E.H.; Saha, A.; Zhu, Z.; Mazurowski, M.A. Breast mass detection in mammography and tomosynthesis via fully convolutional network-based heatmap regression. In *Medical Imaging 2018: Computer-Aided Diagnosis. International Society for Optics and Photonics*; SPIE: Bellingham WA, USA, 2018; Volume 10575, p. 1057525.
181. Dalmış, M.U.; Vreemann, S.; Kooi, T.; Mann, R.M.; Karssemeijer, N.; Gubern-Mérida, A. Fully automated detection of breast cancer in screening MRI using convolutional neural networks. *J. Med Imaging* **2018**, *5*, 014502.
182. Abràmoff, M.D.; Lou, Y.; Erginay, A.; Clarida, W.; Amelon, R.; Folk, J.C.; Niemeijer, M. Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. *Investig. Ophthalmol. Vis. Sci.* **2016**, *57*, 5200–5206.
183. Winkels, M.; Cohen, T.S. 3D G-CNNs for pulmonary nodule detection. *arXiv* **2018**, arXiv:1804.04656.
184. Kermany, D.S.; Goldbaum, M.; Cai, W.; Valentim, C.C.; Liang, H.; Baxter, S.L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell* **2018**, *172*, 1122–1131.
185. Food, U. *Drug Administration. FDA Permits Marketing of Artificial Intelligence-Based Device to Detect Certain Diabetes-Related Eye Problems*; SciPol: Durham, NC, USA, 2018.
186. Gutman, D.; Codella, N.C.; Celebi, E.; Helba, B.; Marchetti, M.; Mishra, N.; Halpern, A. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). *arXiv* **2016**, arXiv:1605.01397.

187. González, G.; Ash, S.Y.; Vegas-Sánchez-Ferrero, G.; Onieva Onieva, J.; Rahaghi, F.N.; Ross, J.C.; Díaz, A.; San José Estépar, R.; Washko, G.R. Disease staging and prognosis in smokers using deep learning in chest computed tomography. *Am. J. Respir. Crit. Care Med.* **2018**, *197*, 193–203.
188. Depeursinge, A.; Vargas, A.; Platon, A.; Geissbuhler, A.; Poletti, P.A.; Müller, H. Building a reference multimedia database for interstitial lung diseases. *Comput. Med Imaging Graph.* **2012**, *36*, 227–238.
189. Armato III, S.G.; McLennan, G.; Bidaut, L.; McNitt-Gray, M.F.; Meyer, C.R.; Reeves, A.P.; Zhao, B.; Aberle, D.R.; Henschke, C.I.; Hoffman, E.A.; et al. The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans. *Med Phys.* **2011**, *38*, 915–931.
190. Petersen, R.C.; Aisen, P.; Beckett, L.A.; Donohue, M.; Gamst, A.; Harvey, D.J.; Jack, C.; Jagust, W.; Shaw, L.; Toga, A.; et al. Alzheimer′s disease neuroimaging initiative (ADNI): Clinical characterization. *Neurology* **2010**, *74*, 201–209.
191. Menze, B.H.; Jakab, A.; Bauer, S.; Kalpathy-Cramer, J.; Farahani, K.; Kirby, J.; Burren, Y.; Porz, N.; Slotboom, J.; Wiest, R.; et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med Imaging* **2014**, *34*, 1993–2024.
192. Marcus, D.S.; Wang, T.H.; Parker, J.; Csernansky, J.G.; Morris, J.C.; Buckner, R.L. Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J. Cogn. Neurosci.* **2007**, *19*, 1498–1507.
193. Bowyer, K.; Kopans, D.; Kegelmeyer, W.; Moore, R.; Sallam, M.; Chang, K.; Woods, K. The digital database for screening mammography. In Proceedings of the Third International Workshop on Digital Mammography, Chicago, IL, USA, 9–12 June 1996; Volume 58, p. 27.
194. Suckling, J.; Parker, J.; Dance, D.; Astley, S.; Hutt, I.; Boggis, C.; Ricketts, I.; Stamatakis, E.; Cerneaz, N.; Kok, S.; et al. *Mammographic Image Analysis Society (MIAS) Database v1. 21;* University of Cambridge: Cambridge, UK, 2015.
195. Bandi, P.; Geessink, O.; Manson, Q.; Van Dijk, M.; Balkenhol, M.; Hermsen, M.; Bejnordi, B.E.; Lee, B.; Paeng, K.; Zhong, A.; et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Trans. Med Imaging* **2018**, *38*, 550–560.
196. Moreira, I.C.; Amaral, I.; Domingues, I.; Cardoso, A.; Cardoso, M.J.; Cardoso, J.S. Inbreast: Toward a full-field digital mammographic database. *Acad. Radiol.* **2012**, *19*, 236–248.
197. Staal, J.; Abràmoff, M.D.; Niemeijer, M.; Viergever, M.A.; Van Ginneken, B. Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med Imaging* **2004**, *23*, 501–509.
198. Hoover, A.; Kouznetsova, V.; Goldbaum, M. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Trans. Med Imaging* **2000**, *19*, 203–210.
199. Decencière, E.; Zhang, X.; Cazuguel, G.; Lay, B.; Cochener, B.; Trone, C.; Gain, P.; Ordonez, R.; Massin, P.; Erginay, A.; et al. Feedback on a publicly distributed image database: The Messidor database. *Image Anal. Stereol.* **2014**, *33*, 231–234.
200. Hu, W.; Zhuo, Q.; Zhang, C.; Li, J. Fast branch convolutional neural network for traffic sign recognition. *IEEE Intell. Transp. Syst. Mag.* **2017**, *9*, 114–126.
201. Shao, F.; Wang, X.; Meng, F.; Rui, T.; Wang, D.; Tang, J. Real-time traffic sign detection and recognition method based on simplified Gabor wavelets and CNNs. *Sensors* **2018**, *18*, 3192.
202. Shao, F.; Wang, X.; Meng, F.; Zhu, J.; Wang, D.; Dai, J. Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network. *Sensors* **2019**, *19*, 2288.
203. Cao, J.; Song, C.; Peng, S.; Xiao, F.; Song, S. Improved traffic sign detection and recognition algorithm for intelligent vehicles. *Sensors* **2019**, *19*, 4021.
204. Zhang, J.; Huang, M.; Jin, X.; Li, X. A real-time chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms* **2017**, *10*, 127.
205. Luo, H.; Yang, Y.; Tong, B.; Wu, F.; Fan, B. Traffic sign recognition using a multi-task convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1100–1111.
206. Li, J.; Wang, Z. Real-time traffic sign recognition based on efficient CNNs in the wild. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 975–984.
207. Masood, S.Z.; Shu, G.; Dehghan, A.; Ortiz, E.G. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv* **2017**, arXiv:1703.07330.

208. Laroca, R.; Zanlorensi, L.A.; Gonçalves, G.R.; Todt, E.; Schwartz, W.R.; Menotti, D. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *arXiv* **2019**, arXiv:1909.01754.
209. Chen, R.C.; et al. Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. *Image Vis. Comput.* **2019**, *87*, 47–56.
210. Raza, M.A.; Qi, C.; Asif, M.R.; Khan, M.A. An Adaptive Approach for Multi-National Vehicle License Plate Recognition Using Multi-Level Deep Features and Foreground Polarity Detection Model. *Appl. Sci.* **2020**, *10*, 2165.
211. Gonçalves, G.R.; Diniz, M.A.; Laroca, R.; Menotti, D.; Schwartz, W.R. Real-time automatic license plate recognition through deep multi-task networks. In Proceedings of the 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Paraná, Brazil, 29 October–1 November 2018; pp. 110–117.
212. Arnold, E.; Al-Jarrah, O.Y.; Dianati, M.; Fallah, S.; Oxtoby, D.; Mouzakitis, A. A survey on 3d object detection methods for autonomous driving applications. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3782–3795.
213. Pham, C.C.; Jeon, J.W. Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks. *Signal Process. Image Commun.* **2017**, *53*, 110–122.
214. Li, B.; Zhang, T.; Xia, T. Vehicle detection from 3d lidar using fully convolutional network. *arXiv* **2016**, arXiv:1608.07916.
215. Helbing, D.; Brockmann, D.; Chadefaux, T.; Donnay, K.; Blanke, U.; Woolley-Meza, O.; Moussaid, M.; Johansson, A.; Krause, J.; Schutte, S.; et al. Saving human lives: What complexity science and information systems can contribute. *J. Stat. Phys.* **2015**, *158*, 735–781.
216. Saleh, S.A.M.; Suandi, S.A.; Ibrahim, H. Recent survey on crowd density estimation and counting for visual surveillance. *Eng. Appl. Artif. Intell.* **2015**, *41*, 103–114.
217. Jones, M.J.; Snow, D. Pedestrian detection using boosted features over many frames. In Proceedings of the International Conference on Pattern Recognition, Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
218. Viola, P.; Jones, M.J.; Snow, D. Detecting pedestrians using patterns of motion and appearance. *Int. J. Comput. Vis.* **2005**, *63*, 153–161.
219. Leibe, B.; Seemann, E.; Schiele, B. Pedestrian detection in crowded scenes. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 878–885.
220. Lin, S.F.; Chen, J.Y.; Chao, H.X. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2001**, *31*, 645–654.
221. Junior, J.C.S.J.; Musse, S.R.; Jung, C.R. Crowd analysis using computer vision techniques. *IEEE Signal Process. Mag.* **2010**, *27*, 66–77.
222. Kok, V.J.; Lim, M.K.; Chan, C.S. Crowd behavior analysis: A review where physics meets biology. *Neurocomputing* **2016**, *177*, 342–362.
223. Sun, M.; Zhang, D.; Qian, L.; Shen, Y. Crowd Abnormal Behavior Detection Based on Label Distribution Learning. In Proceedings of the International Conference on Intelligent Computation Technology and Automation, Nanchang, China, 14–15 June 2015; pp. 345–348.
224. Zhao, L.; Li, S. Object Detection Algorithm Based on Improved YOLOv3. *Electronics* **2020**, *9*, 537.
225. Reno, V.; Mosca, N.; Marani, R.; Nitti, M.; D'Orazio, T.; Stella, E. Convolutional neural networks based ball detection in tennis games. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 19–21 June 2018; pp. 1758–1764.
226. Kang, K.; Ouyang, W.; Li, H.; Wang, X. Object detection from video tubelets with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 817–825.
227. Pobar, M.; Ivasic-Kos, M. Active Player Detection in Handball Scenes Based on Activity Measures. *Sensors* **2020**, *20*, 1475.
228. Pobar, M.; Ivašić-Kos, M. Detection of the leading player in handball scenes using Mask R-CNN and STIPS. In Proceedings of the Eleventh International Conference on Machine Vision (ICMV 2018), International Society for Optics and Photonics, Munich, Germany, 1–3 November 2019; Volume 11041, p. 110411V.
229. Pobar, M.; Ivasic-Kos, M. Mask R-CNN and Optical flow based method for detection and marking of handball actions. In Proceedings of the 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, 13–15 October 2018; pp. 1–6.

230. Burić, M.; Pobar, M.; Ivašić-Kos, M. Object detection in sports videos. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 1034–1039.
231. Acuna, D. Towards real-time detection and tracking of basketball players using deep neural networks. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
232. Afif, M.; Ayachi, R.; Said, Y.; Atri, M. Deep Learning Based Application for Indoor Scene Recognition. *Neural Process. Lett.* **2020**, pp. 1–11.
233. Tapu, R.; Mocanu, B.; Zaharia, T. DEEP-SEE: Joint object detection, tracking and recognition with application to visually impaired navigational assistance. *Sensors* **2017**, *17*, 2473.
234. Yang, W.; Tan, R.T.; Feng, J.; Liu, J.; Guo, Z.; Yan, S. Deep joint rain detection and removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1357–1366.
235. Hu, X.; Zhu, L.; Fu, C.W.; Qin, J.; Heng, P.A. Direction-aware spatial context features for shadow detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7454–7462.
236. Yang, Z.; Li, Q.; Wenyin, L.; Lv, J. Shared multi-view data representation for multi-domain event detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**.
237. Hashmi, M.F.; Gupta, V.; Vijay, D.; Rathwa, V. Computer Vision-Based Assistive Technology for Helping Visually Impaired and Blind People Using Deep Learning Framework. In *Handbook of Research on Emerging Trends and Applications of Machine Learning*; IGI Global: Hershey, PA, USA, 2020; pp. 577–598.
238. Buzzelli, M.; Albé, A.; Ciocca, G. A vision-based system for monitoring elderly people at home. *Appl. Sci.* **2020**, *10*, 374.
239. Szegedy, C.; Toshev, A.; Erhan, D. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: San Francisco, CA, USA, 2013; pp. 2553–2561.
240. Du Terrail, J.O.; Jurie, F. On the use of deep neural networks for the detection of small vehicles in ortho-images. In Proceedings of the 2017 IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 4212–4216.
241. 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 Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
242. Erhan, D.; Szegedy, C.; Toshev, A.; Anguelov, D. Scalable object detection using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2147–2154.
243. Ohn-Bar, E.; Trivedi, M.M. Multi-scale volumes for deep object detection and localization. *Pattern Recognit.* **2017**, *61*, 557–572.
244. Huang, C.; He, Z.; Cao, G.; Cao, W. Task-driven progressive part localization for fine-grained object recognition. *IEEE Trans. Multimed.* **2016**, *18*, 2372–2383.
245. Liu, N.; Han, J. DHSNet: Deep hierarchical saliency network for salient object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 678–686.
246. Li, X.; Zhao, L.; Wei, L.; Yang, M.H.; Wu, F.; Zhuang, Y.; Ling, H.; Wang, J. DeepSaliency: Multi-task deep neural network model for salient object detection. *IEEE Trans. Image Process.* **2016**, *25*, 3919–3930.
247. Wang, L.; Lu, H.; Ruan, X.; Yang, M.H. Deep networks for saliency detection via local estimation and global search. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3183–3192.
248. Li, G.; Yu, Y. Deep contrast learning for salient object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 26 June–1 July 2016; pp. 478–487.
249. Gao, M.L.; He, X.; Luo, D.; Yu, Y.M. Object tracking based on harmony search: Comparative study. *J. Electron. Imaging* **2012**, *21*, 043001.
250. Hao, Z. Improved Faster R-CNN for Detecting Small Objects and Occluded Objects in Electron Microscope Imaging. *Acta Microsc.* **2020**, *29*.

251. Leung, H.K.; Chen, X.Z.; Yu, C.W.; Liang, H.Y.; Wu, J.Y.; Chen, Y.L. A Deep-Learning-Based Vehicle Detection Approach for Insufficient and Nighttime Illumination Conditions. *Appl. Sci.* **2019**, *9*, 4769.
252. Park, J.; Chen, J.; Cho, Y.K.; Kang, D.Y.; Son, B.J. CNN-based person detection using infrared images for night-time intrusion warning systems. *Sensors* **2020**, *20*, 34.
253. Kim, K.H.; Hong, S.; Roh, B.; Cheon, Y.; Park, M. PVANET: Deep but lightweight neural networks for real-time object detection. *arXiv* **2016**, arXiv:1608.08021.
254. Shih, Y.F.; Yeh, Y.M.; Lin, Y.Y.; Weng, M.F.; Lu, Y.C.; Chuang, Y.Y. Deep co-occurrence feature learning for visual object recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21–26 July 2017; pp. 4123–4132.
255. Denton, E.L.; Chintala, S.; Szlam, A.; Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montréal, ON, Canada, 7–12 December 2015; Volume 1, pp. 1486–1494.
256. Takác, M.; Bijral, A.S.; Richtárik, P.; Srebro, N. Mini-Batch Primal and Dual Methods for SVMs. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1022–1030.
257. Goring, C.; Rodner, E.; Freytag, A.; Denzler, J. Nonparametric part transfer for fine-grained recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, USA, 24–27 June 2014; pp. 2489–2496.
258. Lin, D.; Shen, X.; Lu, C.; Jia, J. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1666–1674.
259. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-based R-CNNs for fine-grained category detection. In Proceedings of the European Conference on Computer Vision, Zürich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014; pp. 834–849.
260. RaspberryPI. Available online: https://www.raspberrypi.org/ (accessed on 31 December 2019).
261. Nakahara, H.; Yonekawa, H.; Sato, S. An object detector based on multiscale sliding window search using a fully pipelined binarized CNN on an FPGA. In Proceedings of the International Conference on Field Programmable Technology, Melbourne, Australia, 11–13 December 2017; pp. 168–175.
262. Soma, P.; Jatoth, R.K. Hardware Implementation Issues on Image Processing Algorithms. In Proceedings of the International Conference on Computing Communication and Automation, India, 14–15 December 2018; pp. 1–6.
263. JetsonTX2. Available online: https://elinux.org/JetsonTX2 (accessed on 31 December 2019).
264. Garland, M.; Le Grand, S.; Nickolls, J.; Anderson, J.; Hardwick, J.; Morton, S.; Phillips, E.; Zhang, Y.; Volkov, V. Parallel computing experiences with CUDA. *IEEE Micro* **2008**, *28*, 13–27.
265. Stone, J.E.; Gohara, D.; Shi, G. OpenCL: A parallel programming standard for heterogeneous computing systems. *Comput. Sci. Eng.* **2010**, *12*, 66–73.
266. NVIDIA Collective Communications Library (NCCL). Available online: https://developer.nvidia.com/nccl (accessed on 31 December 2019).
267. Hwang, S.; Lee, Y. FPGA-based real-time lane detection for advanced driver assistance systems. In Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems, Jeju, South Korea, 25–28 October 2016; pp. 218–219.
268. Sajjanar, S.; Mankani, S.K.; Dongrekar, P.R.; Kumar, N.S.; Mohana.; Aradhya, H.V.R. Implementation of real time moving object detection and tracking on FPGA for video surveillance applications. In Proceedings of the IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 13–14 August 2016; pp. 289–295.
269. Tijtgat, N.; Van Ranst, W.; Goedeme, T.; Volckaert, B.; De Turck, F. Embedded real-time object detection for a UAV warning system. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2110–2118.
270. Hossain, S.; Lee, D.j. Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices. *Sensors* **2019**, *19*, 3371.
271. Stepanenko, S.; Yakimov, P. Using high-performance deep learning platform to accelerate object detection. In Proceedings of the International Conference on "Information Technology and Nanotechnology, Samara, Russia, 26–29 May 2019; pp. 1–7.

272. Körez, A.; Barışçı, N. Object Detection with Low Capacity GPU Systems Using Improved Faster R-CNN. *Appl. Sci.* **2020**, *10*, 83.
273. Çambay, V.Y.; Uçar, A.; Arserim, M.A. Object Detection on FPGAs and GPUs by Using Accelerated Deep Learning. In Proceedings of the 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 28–30 September 2019; pp. 1–5.
274. Moon, Y.Y.; Geem, Z.W.; Han, G.T. Vanishing point detection for self-driving car using harmony search algorithm. *Swarm Evol. Comput.* **2018**, *41*, 111–119.
275. Yao, Y.; Wang, Y.; Guo, Y.; Lin, J.; Qin, H.; Yan, J. Cross-dataset Training for Class Increasing Object Detection. *arXiv* **2020**, arXiv:2001.04621.
276. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.