future internet

Article

NextDet: Efficient Sparse-to-Dense Object Detection with Attentive Feature Aggregation

Priyank Kalgaonkar and Mohamed El-Sharkawy *

Department of Electrical and Computer Engineering, Purdue School of Engineering and Technology Indianapolis, Indianapolis, IN 46254, USA
* Correspondence: melshark@purdue.edu

Abstract: Object detection is a computer vision task of detecting instances of objects of a certain class, identifying types of objects, determining its location, and accurately labelling them in an input image or a video. The scope of the work presented within this paper proposes a modern object detection network called NextDet to efficiently detect objects of multiple classes which utilizes CondenseNeXt, an award-winning lightweight image classification convolutional neural network algorithm with reduced number of FLOPs and parameters as the backbone, to efficiently extract and aggregate image features at different granularities in addition to other novel and modified strategies such as attentive feature aggregation in the head, to perform object detection and draw bounding boxes around the detected objects. Extensive experiments and ablation tests, as outlined in this paper, are performed on Argoverse-HD and COCO datasets, which provide numerous temporarily sparse to dense annotated images, demonstrate that the proposed object detection algorithm with CondenseNeXt as the backbone result in an increase in mean Average Precision (mAP) performance and interpretability on Argoverse-HD's monocular ego-vehicle camera captured scenarios by up to 17.39% as well as COCO's large set of images of everyday scenes of real-world common objects by up to 14.62%.

Keywords: CondenseNeXt; object detection; PyTorch; deep learning; convolutional neural network

Citation: Kalgaonkar, P.; El-Sharkawy, M. NextDet: Efficient Sparse-to-Dense Object Detection with Attentive Feature Aggregation. *Future Internet* **2022**, *14*, 355. https://doi.org/10.3390/fi14120355

Academic Editor: Manuel José Cabral dos Santos Reis

Received: 30 September 2022
Accepted: 23 November 2022
Published: 28 November 2022

1. Introduction

Artificial Intelligence (AI) is a branch of computer science that is widely studied by scientists and engineers to build machines to mimic human intelligence. Driven by curiosity and demand, humans have been trying to infuse AI into modern machines built to perform labor-intensive tasks. Mainstream research fields related to AI include, but not limited to, computer vision and pattern recognition, robotics and automation, natural language processing and their current real-world applications include Amazon Rekognition, an automated image and video analysis tool, to Tesla's Full Self-Driving software [1,2].

Deep Convolutional Neural Network (Deep CNN) is a subset of Deep Neural Network (DNN) popular in the field of AI and commonly used for designing innovative techniques and algorithms for computer vision systems to perform complex operations such as image classification and object detection using multiple layers of neurons to simulate natural vision perception of human beings. Practical applications such as autonomous vehicles and autonomous robots using edge devices require accuracy and inference times close to that of human visual system. Existing real-time object detectors which provide higher accuracy, do not operate in real-time and require multiple Graphics Processing Units (GPU) to train the network on large datasets [3], which has fueled a need to develop efficient algorithms and innovative techniques to detect objects and perceive the surrounding environment using computer vision.

In this paper, we propose a modern object detector called NextDet, with a goal to improve overall object detection accuracy whilst maintaining computational efficiency of

CondenseNeXt [4], a light-weight CNN, which is designed to reduce number of Floating-Point Operations (FLOPs) and parameters. The work presented within this paper can be summarized into three points as follows:

1. An efficient and a powerful object detector is being proposed which can be trained on a single GPU.
2. State-of-the-art algorithms and methodologies such as FPN [5], PAN [6], YOLO [7], GIoU_Loss [8], etc. have been modified and implemented into the design of the proposed object detector.
3. Extensive training and evaluation experiments have been performed on two large-scale multi-class datasets: Argoverse-HD [9] and Microsoft COCO [10].
4. The remainder of this paper has been partitioned as follows: Section 2 provides the reader a detailed information about object detection models in general and its related works. Section 3 defines the methodology of the proposed NextDet object detection network. Section 4 evaluates the NextDet object detector by benchmarking on different datasets and Section 5 concludes this paper.

2. Anatomy of Object Detectors and Related Works

A modern object detector is tasked to determine *where* and *what* type of object is in each input image. It is composed of three main parts: a *backbone* to extract features to provide a feature map representation of the input image using a robust image classifier, a *neck* linked to the backbone which acts like a feature aggregator by collecting feature maps from different stages of the backbone and fuses these multi-level features together, and a *head* to determine bounding boxes and perform class predictions. Figure 1 provides a visual representation of backbone, neck, and head of a modern object detector.

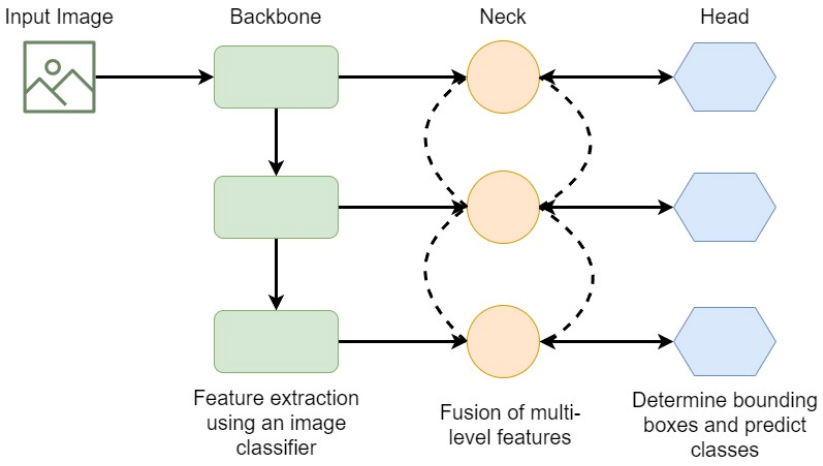

Figure 1. Backbone, neck, and head modules of a modern object detector. The backbone module utilizes a robust image classifier to extract features from an input image's multiple resolutions and the neck, which is linked to the backbone, acts like a feature aggregator to fuse features of multiple resolutions. The head module then determines bounding boxes and predicts classes.

An object detector's head is further classified into two main types: one-stage object detectors (dense predictions) such as You Only Look Once (YOLO) [3], Single Shot Detector (SSD) [11], RetinaNet [12], CornerNet [13], and CenterNet [14], and two-stage object detectors (sparse predictions) such as R-FCN [15], Relation Network [16] and family of R-CNN [17] namely Fast R-CNN [18], Faster R-CNN [19], Mask-RCNN [20], Cascade R-CNN [21] and Libra R-CNN [22].

Two-stage detectors such as the family of R-CNN perform these tasks separately in different modules by utilizing a Region Proposal Network in the first stage to generate sparse region proposals to obtain Region of Interest (RoI) and then passing down these region proposals for object classification and bounding-box regressions in the second stage resulting in an increase in object detection accuracy but leading to a time complexity bottle-

neck and therefore, an increase in demand of computational resources for implementation purposes. On the other hand, one-stage detectors such as YOLO and SSD execute image classification and object detection tasks in a single module but may not be able to achieve a desirable real-time inference performance due to constrained computational resources on edge devices. In comparison, YOLO obtains a better inference performance at the cost of detection performance whereas SSD obtains a better detection performance at the cost of inference performance [23]. State-of-the-art YOLO algorithm is a popular choice in designing modern object detectors because of its speed and accuracy performance. Figure 2 provides a visual comparison between a one-stage object detector and a two-stage object detector.

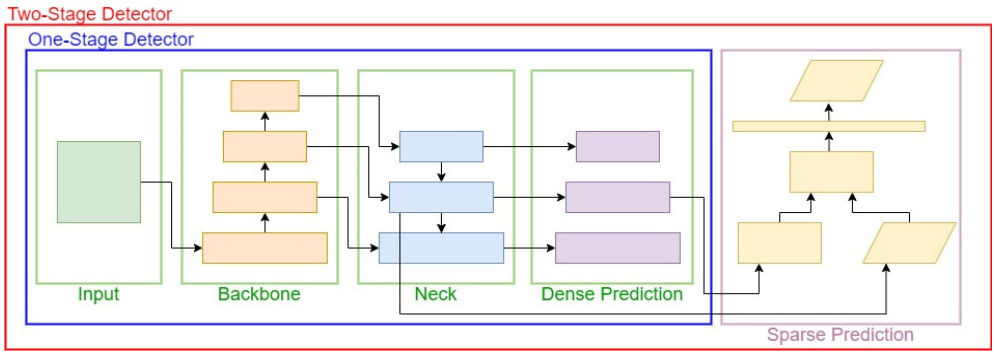

Figure 2. A visual comparison between a one-stage object detector vs. a two-stage object detector.

An object detector's neck is comprised of multiple top-down and bottom-up paths linked to the backbone of the detection network to facilitate fusion of multi-level features. Popular techniques implemented in the neck of the network include incorporating additional blocks such as Spatial Pyramid Pooling [24] and/or Receptive Field Block [25] or utilizing path-aggregation blocks such as Feature Pyramid Networks [5], Path Aggregation Network [6] and/or adaptively spatial feature fusion (ASFF) [26], to name a few. The backbone architecture is usually implemented using a robust image classifier as shall be seen in Section 3.

In general, a modern object detector model is made up of a combination of the following techniques within each module:

- **Input:** Images
- **Backbone:** A robust image classifier (CNN) such as CondenseNeXt [4], CondenseNet [27], MobileNetv3 [28], VGG16 [29], ResNet-50 [30], SpineNet [31]
- **Neck:**
  - **Additional Blocks:** SPP [24], RFB [25], ASPP [32]
  - **Path Aggregation Blocks:** FPN [5], PAN [6], ASFF [26]
- **Head:**
  - **One-Stage Predictions (Dense):** YOLO [7], SSD [11], RetinaNet [12], CornerNet [13], CenterNet [14]
  - **Two-Stage Predictions (Sparse):** R-FCN [15], Relation Network [16], R-CNN [17], Fast R-CNN [18], Faster R-CNN [19], R-FCN [15], Mask-RCNN [20], Cascade R-CNN [21], Libra R-CNN [22]

3. NextDet

This paper introduces a modern object detection network, NextDet, built upon YOLOv5 [33] object detection framework for efficient monocular sparse-to-dense streaming perception, especially for autonomous vehicles and autonomous rovers using edge devices. NextDet is faster, lighter and can perform both, sparse and dense object detection efficiently, as seen from experiments performed on different datasets in Section 4, from which results are extrapolated accordingly.

3.1. Backbone

A backbone of a modern object classifier incorporates a robust image classifier to extrapolate essential features from an input image, at different levels of coarseness. For devices with constrained computational resources, it is crucial to consider computational performance along with prediction accuracy of the image classifier. The versions of the officially published YOLO [7] family utilize Darknet53 architecture as the backbone of the network which results in an improved detection performance at the cost of heavy parametrization.

NextDet incorporates a modified version of the CondenseNeXt [4] CNN architecture as the backbone of the proposed object detection network. CondenseNeXt belongs to the family of DenseNet [34] which provides a novel technique of extracting spatial information from each layer and forwarding it to every other subsequent layer in a feed-forward fashion, allowing this information to be extracted at different levels of coarseness. CondenseNeXt utilizes multiple dense blocks at its core, as seen in Figure 3, and depthwise separable convolution and pooling layers in between these blocks to change feature-map sizes so that features can be extrapolated more efficiently at different resolutions from an input image and then fuse this information to address the vanishing-gradient problem.

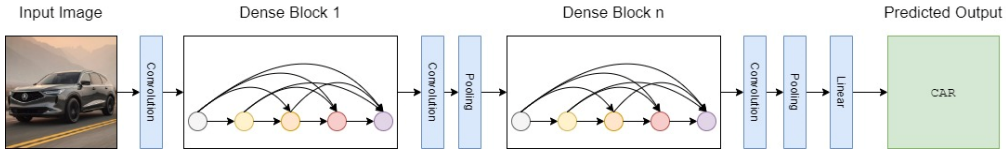

Figure 3. A general visual representation of convolution and pooling layers used to change feature map sizes between multiple (n) dense blocks of CondenseNeXt architecture which allows it to efficiently extract information of features from an input image.

CondenseNeXt backbone utilized in the proposed object detection network, NextDet, is re-engineering to be furthermore lightweight than its initial design for image classification purposes, by deprecating final layers of classification in order to make it compatible and to allow to link the backbone to the neck module as discussed in Section 3.1.1 and visually represented in Figure 4.

3.1.1. CondenseNeXt CNN

NextDet is designed with CondenseNeXt [4] CNN as the backbone. It is an efficient, yet robust image classification network designed to reduce the amount of computational resources required for real-time inference on resource constrained computing systems. It utilizes depthwise separable convolution wherein, a depthwise convolution layer applies 3×3 depthwise convolution $\hat{K}$ to a single input channel instead of all input channels mathematically represented by Equation (1), and then a 1×1 pointwise convolution $\widetilde{K}$ performs a linear combination of all the output obtained from depthwise convolution mathematically represented by Equation (2) as follows:

$$\hat{Y}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot X_{k+1-1,l+j-1,m} \tag{1}$$

$$Y_{k,l,n} = \sum_{m} \widetilde{K}_{m,n} \cdot \hat{Y}_{k-1,l-1,m} \tag{2}$$

Here, X represents input feature map of size $D_x \times D_x \times I$ and Y represents output feature map of size $D_x \times D_x \times O$ where I corresponds to the number of input channels and O corresponds to the number of output channels. Figure 5 provides a graphical representation of the depthwise separable convolution.

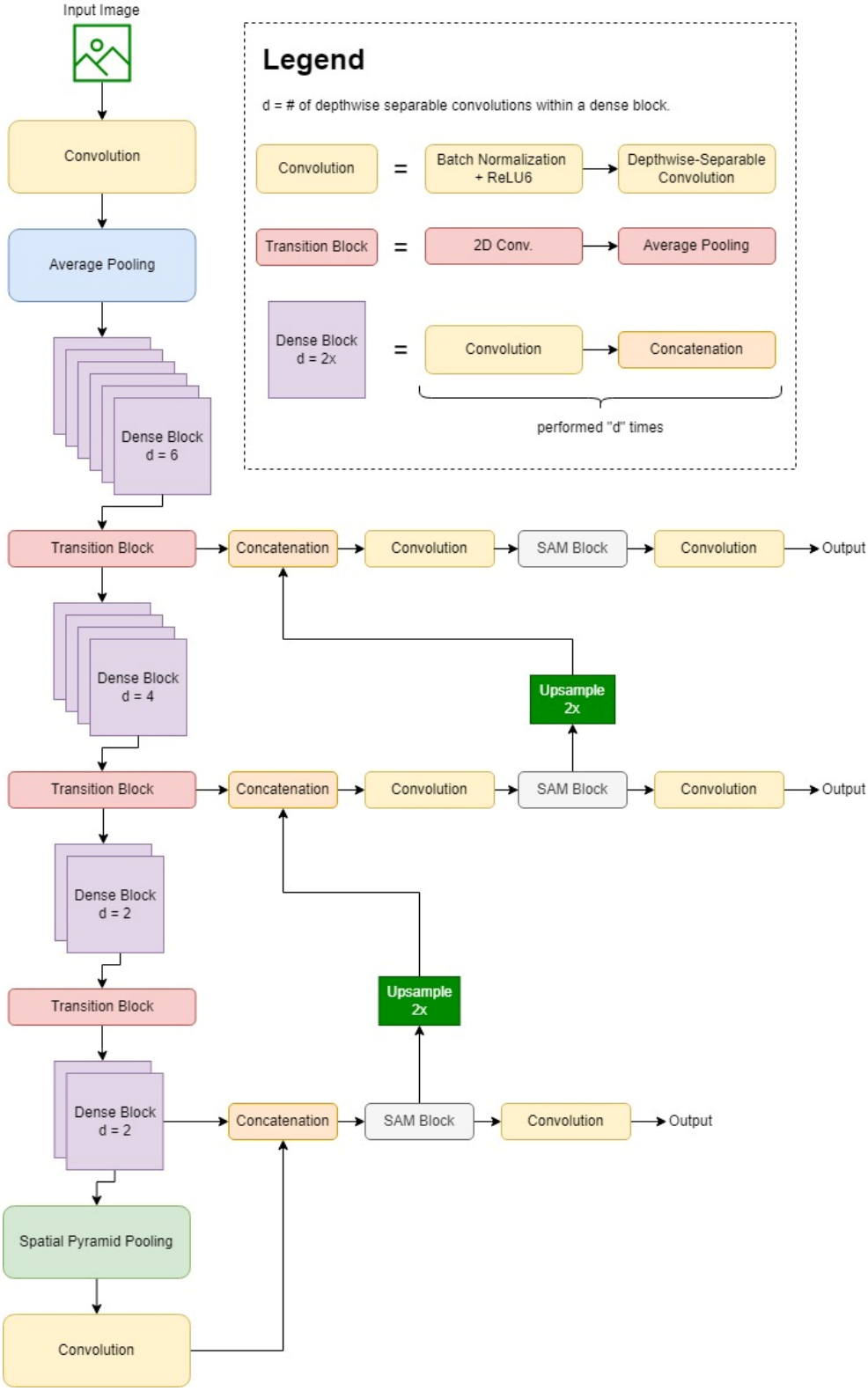

Figure 4. A visual representation of the overall mechanism of the proposed NextDet object detection network. The *backbone* of this proposed network consists of CondenseNeXt image classifier, the *neck* consists of Feature Pyramid Network (FPN) with Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN) methodologies, and the *head* consists of an attention block and YOLO pipeline.

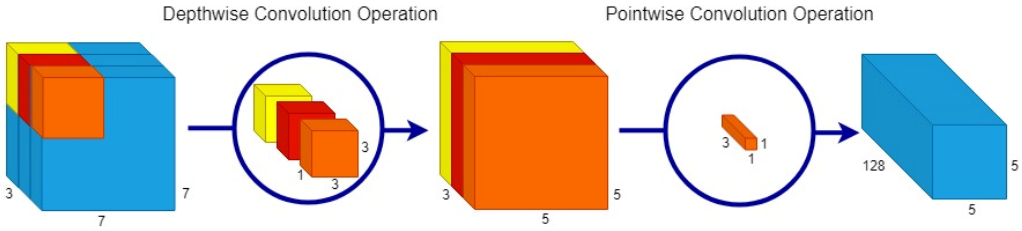

Figure 5. A 3D representation of the efficient Depthwise Separable Convolution (DSC) operation. In this image, DSC operation performs a 3×3 depthwise convolution operation on an input image which transforms the image and then performs a 1×1 pointwise convolution which elongates this transformed image to over 128 channels. For example, a blue (B) channel from RGB image is elongated only one to obtain different shades of blue channel, instead of transforming the same image multiple times.

The design of CondenseNeXt also incorporates group-wise pruning to prune and to exorcise inconsequential filters before the first stage of the depthwise separable convolution during the training process which is arbitrated by L1-Normalization and balanced focal loss function techniques to smoothen the harsh effects of the pruning process. Equation (3) defines the count of inconsequential filters exorcised during this pruning stage, where G denotes group convolution, C denotes cardinality and p is a hyper-parameter defined for the training process to indicate a desired number of filters to be pruned, as follows:

$$G \cdot C_x = I \cdot C - p \cdot I \tag{3}$$

A new dimension is added into the design of this CNN, called Cardinality, to help improve and recover accuracy lost during the filter pruning process. To help improve the classification accuracy performance even further, ReLU6 (Rectified Linear Units capped at 6) [35] activation function has been implemented into the design by allowing the network to learning sparse features earlier than it would using ReLU activation function. ReLU6 also addresses ReLU's exploding gradient problem [36]. ReLU6 can be mathematically defined as follows:

$$f(x) = min(max(0, x), 6) \tag{4}$$

CondenseNeXt utilizes multiple dense blocks at its core where ReLU6 activation function, Batch Normalization (BN) and convolution blocks sequentially constitute a single dense block, as seen in Figure 4. Such a dense block with n number of convolutions can be defined by the following operation where y_i denotes feature maps obtained from a $A_i(\cdot)$ non-linear function, as follows:

$$y_n = A_n([x_0\ ,\ x_1\ ,\ x_2\ ,\ x_3\ ,\ \cdots\ ,\ x_{n-1}]) \tag{5}$$

3.2. Neck

3.2.1. FPN and PAN

The neck module of a modern object detector is linked to the backbone module which acts like a feature aggregator by collecting feature maps from different stages of the backbone and fusing them together with the help of pyramid networks such as Feature Pyramid Networks (FPN) [5] and Path Aggregation Network (PANet or PAN) [6]. The neck of the proposed object detector is implemented by connecting the PAN to the FPN. The FPN provides feature maps of different sizes in order to fuse different features together. However, since the feature maps are of different sizes in the feature pyramid, the bottom features cannot be fused with the features on the top. To address this issue, the PAN is connected to the FPN and up sampled by a factor of two using the nearest neighbor approach, allowing bottom features to be connected to the top features. The bottom-up approach provides strong positioning features, and the top-down approach provides strong

semantic features, resulting in an improvement in object detection performance. Figure 6 provides a simple graphical representation of this technique.

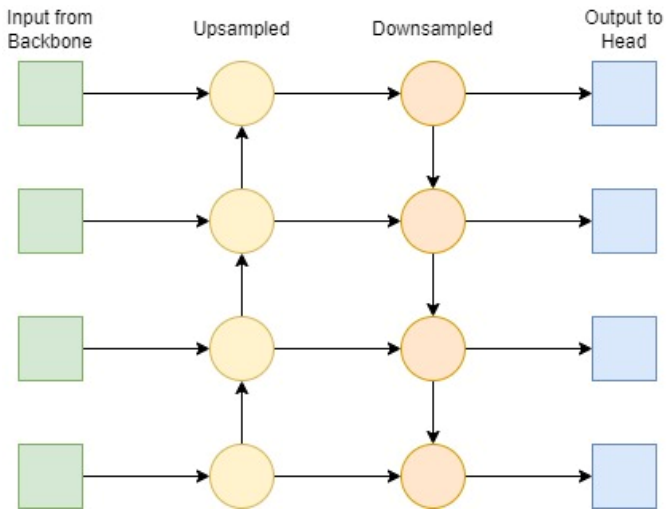

Figure 6. A graphical representation of the PANet implementation of neck of the proposed NextDet.

3.2.2. SPP

Spatial Pyramid Pooling (SPP) [24] is a novel pooling strategy to boost recognition accuracy performance of CNNs by pooling responses of every filter in each local spatial bin and maintaining this spatial information, inspired by the famous Bag of Words [37] approach in computer vision. This approach utilizes three different sizes of max-pooling operations in order to identify analogous feature maps irrespective of varying resolutions of input feature patterns. The output from these max-pooling operations is then flattened, concatenated, and sent to the Fully Connected (FC) layer with fixed-size output irrespective of the input size, as seen in Figure 7. The fixed-size output constraint of the CNNs is due to the FC layer, not the convolution layer. Hence, in the design of the proposed object detection network, SPP is implemented in the final stages of feature extraction, as seen in Figure 4.

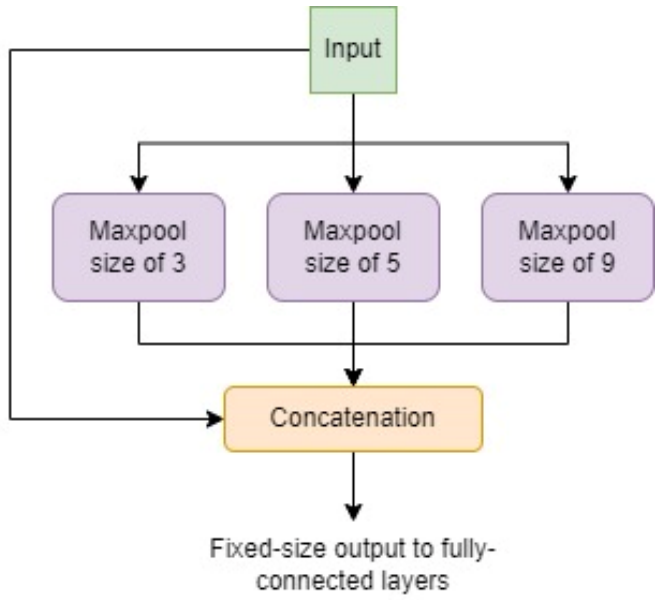

Figure 7. A graphical representation of the SPP block in the neck of the proposed NextDet object detector.

3.3. Head

The head module of a modern object detector determines bounding boxes and outputs detection results such as object class, score, location, and size. The general design of such systems incorporates multiple head modules in order to precisely detect objects in an input image and predict scores by exploiting a common feature set from earlier in the network. The design of the proposed NextDet object detection network utilizes 3 heads using the YOLO layer as seen in Figure 4.

Each head of the proposed NextDet network incorporates a highly efficient Spatial Attention Module (SAM) block, introduced in Convolutional Block Attention Module (CBAM) [38], for attentive feature aggregation. In this approach, a spatial attention map is generated with a focus on more important parts. Average-pooling and max-pooling operations along the channel axis is utilized to obtain inter-spatial relationships of features and can be defined mathematically as follows:

$$y = \sigma\Big(N_f(x)\Big) \times x \tag{6}$$

Here, x denotes the input feature map, y denotes the output feature map and N_f denotes non-linear function for computing a SAM block's output. $\sigma\Big(N_f(x)\Big)$ corresponds to the attention function which assigns a value up to 1 for spatial features of a higher priority and a value down to 0 for spatial features of a lower priority of the input x.

3.4. Bounding Box Regression

The complex task of object detection can be divided into following two sub-tasks for simplicity: object classification and object localization. The task of object localization depends on Bounding Box Regression (BBR) to locate an object of interest within an image and draw a predicted rectangular bounding box around it by overlapping the area within the predicted bounding box and the ground truth bounding box. This overlap area is called as Intersection over Union (IoU) loss. IoU, also known as Jaccard Index, is a popular metric used to measure diversity and similarity of two arbitrary shapes. It can be defined as the ratio of the intersection and union of the predicted bounding box (A) and the ground-truth bounding box (B), and can be mathematically defined as follows:

$$IoU(A,\ B) = \frac{A\ \cap\ B}{A\ \cup\ B} \tag{7}$$

IoU loss is a popular BBR approach, but it does not work when predicted bounding box and the ground truth bounding box do not overlap i.e., when $IoU(A,\ B) = 0$. To overcome this disjoint condition of A and B in IoU, Generalized IoU (GIoU) [8] loss has been implemented into the design of the proposed NextDet object detector. GIoU solves IoU's disjoint problem by increasing the overlap area size between the predicted bounding box and the ground truth bounding box, and slowly moving the predicted bounding box towards the target ground truth box. GIoU loss can be mathematically represented by Equation (8) as follows:

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - A \cup\ B|}{|C|} \tag{8}$$

Here, C is the smallest box covering A and B. Experiments in [8] demonstrate that GIoU achieves an improved performance over Mean Squared Error (MSE) and IoU losses. Furthermore, the experiments demonstrate that GIoU is effective in addressing vanishing gradients in non-overlapping cases.

4. Experimental Analysis

A modern object detector not only determines where the object is in an image but also determines what type of object it is by drawing rectangular bounding boxes around the object in consideration. Such an object detector is usually trained on a dataset contained

labelled images, also known as ground-truth values. These ground-truth values provide pre-labelled images with class and object labels along with bounding boxes for each object within an image.

4.1. Datasets

Robustness of the proposed modern object detector, NextDet, is evaluated on two publicly available multi-class datasets: Argoverse-HD [9] and Microsoft COCO [10]. These datasets provide annotated images and bounding boxes for object detection, tracking and forecasting for autonomous driving, everyday objects, and humans. Each dataset is split into a pre-defined train and test sets where each set contains a fixed number of images. Figure 8 provides an overview of images from each of the two datasets utilized to train and test the proposed object detector. Table 1 provides a statistical overview and comparison between these two datasets. The two datasets chosen possess images with unique features and characteristics, as seen in Table 1 and Figure 8, in order to provide an extensive evaluation of the proposed object detector.

(**a**) (**b**)

(**c**) (**d**)

Figure 8. An overview of raw images taken from the two datasets used for training and testing the proposed object detector. Images (**a**,**b**) denote sparse and dense multi-class objects from the Argoverse-HD dataset in overcast and clear conditions respectively, and images (**c**,**d**) denote sparse and dense multi-class objects from the COCO dataset in grayscale and RGB color space respectively. These datasets contain multiple images containing objects of different classes, complex scenarios, and diverse lighting.

Table 1. Statistics of the two datasets utilized in training and testing the proposed object detector.

Dataset	Type	Number of Classes	Number of Images		Resolution	Class Overlap
			Train	Test		
Argoverse-HD	Road	8	39,384	15,062	1200 × 1920	True
COCO	Common	80 [1]	118,287	5000	Multi-scale	True

[1] COCO dataset features 80 object classes and an additional 11 *stuff* categories with missing object categories/labels.

4.1.1. Argoverse-HD

Argoverse-HD [9] dataset, where HD stands for High-framerate Detection, was introduced for Streaming Perception Challenge competition in August 2020 and is a derivative of Argoverse 1.1 dataset [39]. It offers diverse urban outdoor scenes composed of 204 linear kilometers in the city of Miami and 86 linear kilometers in the city of Pittsburgh using a high frame-rate camera sensor data at 30 FPS.

To satisfy the purpose of monocular sparse-to-dense streaming perception evaluation of the proposed object detector, only images from the center ring RGB camera of the Argoverse 1.1′s ego-vehicular camera setup is utilized for training and testing purposes. This dataset contains 8 annotated object classes related to common world driving scenarios such as bicycle, bus, car, motorcycle, person, stop sign, traffic light, and truck with a total of 66,953 images of a fixed 1200 × 1920 resolution and 1.26 million bounding boxes.

4.1.2. COCO

Common Objects in Context (COCO) [10] is a popular large-scale object detection, segmentation, and captioning dataset introduced by Microsoft in 2014 containing 328,000 multi-scale resolution images and 2.5 million labeled instances of everyday scenes of common objects in their natural environment.

To increase the diversity in training and evaluation of the proposed object detector, COCO dataset was chosen. Some of the object classes including all of the 8 object classes from the Argoverse-HD dataset are person, bicycle, car, motorcycle, airplane, bus, horse, sheep, cow, and train, to a name a few. In total, COCO dataset offers 80 object classes and an additional 11 *stuff* object classes with missing object labels and no clear boundaries such as grass, street, sky, etc. also provide significant contextual information. It, therefore, provides numerous challenging scenarios to further evaluate the performance of the proposed object detector.

4.2. Model Evaluation Metrics

An object detector is assumed to have detected target objects successfully upon fulfilling the following two constraints. Using a grid search approach, IoU threshold of 0.2 is selected and utilized throughout all models and datasets outlined in this paper. A two-step evaluation procedure is performed to evaluate the proposed object detector, NextDet as follows:

1. Calculating the predicted bounding box and the ground-truth bounding box ratio: IoU defined in (7) is utilized to calculate this ratio of overlap. If the calculated IoU value is greater than a pre-determined threshold, it is determined that the object detector has detected the target object within an image successfully.
2. Matching the ground-truth and predicted bounding box class labels: After it is determined that an object has been successfully detected in step 1, class label of the predicted bounding box is matched to the ground-truth bounding box accordingly.

Different metrics are used to evaluate an object detector [40]. The Mean Average Precision (mAP) is a popular evaluation and benchmarking metric defined as the average of the Average Precision (AP) calculated for all classes. AP corresponds the area under the Precision-Recall (PR) curve where Precision (P) measures the ability for a model to identify

target objects i.e., the percentage of predictions that are correct, and Recall (R) measures the ability a model to find all positives among all relevant ground truths. These metrics can be defined mathematically as follows:

$$P = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{9}$$

$$R = \frac{TP}{TP + FN} = \frac{TP}{all\ detections} \tag{10}$$

$$AP = \frac{\Sigma P}{number\ of\ objects} \tag{11}$$

$$mAP = \frac{\Sigma AP}{number\ of\ classes} \tag{12}$$

Here, True Positives (TP) denotes a correct detection when IoU value is greater than or equal to the threshold usually set between 50% to 95%, False Positives (FP) denotes a wrong detection when IoU value is smaller than the threshold, and False Negative (FN) indicates ground truth not being detected.

4.3. Experiment Setup

The proposed object detector, NextDet, is implemented using the Python programming language, and has been designed and developed using the latest version of the PyTorch [41] framework (at the time of paper submission). For training, a batch size of 8 and cosine decay is used in order to facilitate resetting of the learning rate after each epoch, for 300 epochs.

Single-GPU training for experiments presented within this paper were performed on one node of Carbonate supercomputer's GPU partition, a large-memory computer cluster with 24 GPU-accelerated Apollo 6500 nodes, specialized for deep learning research. Each node of the Carbonate's GPU partition is provided and managed by the Research Technologies division at the Indiana University, which supported the work presented within this paper, in part by Shared University Research grants from IBM Inc. to Indiana University and Lilly Endowment Inc. through its support for the Indiana University Pervasive Technology Institute [42], and consists of:

- NVIDIA Tesla V100 PCIe 32 GB GPU
- Intel 6248 2.5 GHz 20-core CPU
- 1.92 TB solid-state drive
- 768 GB of RAM
- PyTorch 1.12.1
- Python 3.7.9
- CUDA 11.3

4.4. Experiment Results

Ablation studies are crucial for any deep learning research and in this paper, an extensive analysis is performed on a single GPU by removing and/or replacing certain parts from the proposed complex deep neural network in order to better understand the network's performance. Six different versions of the proposed NextDet network have been created, trained, and evaluated on two aforementioned datasets to determine the importance of each module

Furthermore, to test robustness and obtain a fair comparison result, other PyTorch versions of lightweight networks such as ShuffleNet [43] and MobileNetv3 [28] are attached as backbone with NextDet and benchmarked in Tables 2 and 3. ShuffDet + PAN and Mob3Det + PAN denotes ShuffleNet and MobileNetv3 CNNs with the proposed FPN and PAN networks, ShuffDet + FPN and Mob3Det + FPN denotes ShuffleNet and MobileNetv3 CNNs with only FPN in its neck design respectively. Table 4 provides a comparison of image classification performance of the backbones [4,28,43] in terms of floating-point

operation and number of trainable parameters, and Table 5 provides a description of each version of the object detector used in this experimental analysis utilized in experiment analyses described in Tables 2 and 3.

Table 2. A summary of performance comparison benchmarked on the Argoverse dataset.

Architecture Name	Backbone	Mean Precision	mAP	AP_{50}	AP_{75}
NextDet	CondenseNeXt	61.90%	35.10%	48.95%	38.20%
NextDet + FPN	CondenseNeXt	58.06%	33.88%	45.16%	36.91%
NextDet-SAM	CondenseNeXt	59.70%	34.49%	45.82%	37.20%
ShuffDet + PAN	ShuffleNet	58.31%	30.30%	42.21%	32.95%
ShuffDet + FPN	ShuffleNet	53.95%	28.97%	40.80%	31.54%
Mob3Det + PAN	MobileNetv3	54.80%	29.90%	41.57%	32.53%
Mob3Det + FPN	MobileNetv3	52.97%	28.56%	40.09%	31.05%

Table 3. A summary of performance comparison benchmarked on the Microsoft COCO dataset.

Architecture Name	Backbone	Mean Precision	mAP	AP_{50}	AP_{75}
NextDet	CondenseNeXt	66.80%	58.00%	81.63%	63.73%
NextDet + FPN	CondenseNeXt	64.86%	56.76%	79.65%	61.87%
NextDet-SAM	CondenseNeXt	65.36%	57.14%	80.41%	62.27%
ShuffDet + PAN	ShuffleNet	59.40%	51.10%	72.40%	56.33%
ShuffDet + FPN	ShuffleNet	57.51%	50.78%	71.02%	55.00%
Mob3Det + PAN	MobileNetv3	54.50%	50.60%	70.95%	54.69%
Mob3Det + FPN	MobileNetv3	53.83%	50.12%	70.18%	54.23%

Table 4. Comparison of light-weighted CNN backbone performance with a scaling factor of 0.5 on CIFAR-10 dataset.

Light-Weight CNN	FLOPs [1]	Parameters	Top-1 Accuracy [2]
CondenseNeXt	26.8 million	0.16 million	92.28%
ShuffleNet	43.43 million	0.25 million	91.48%
MobileNetv3	36.34 million	1.84 million	88.93%

[1] FLOPs: The number of floating-point operations to measure computational complexity. [2] Top-1 Accuracy: Indicates correct prediction rate for a class with highest probability.

Table 5. A summary of acronyms of different versions of object detection architectures utilized in training and analysis phase.

Architecture Name	Description
NextDet	The proposed NextDet architecture in this paper.
NextDet + FPN	NextDet architecture with FPN network only.
NextDet-SAM	NextDet architecture without spatial attention modules.
ShuffDet + PAN	NextDet architecture with ShuffleNet backbone and PAN network.
ShuffDet + FPN	NextDet architecture with ShuffleNet backbone and FPN network.
Mob3Det + PAN	NextDet architecture with MobileNetv3 backbone and PAN network.
Mob3Det + FPN	NextDet architecture with MobileNetv3 backbone and FPN network.

Extrapolating benchmarking results from Tables 2 and 3, it can be observed that the proposed object detector, NextDet, achieves a notable performance in Argoverse's monocular streaming perception as well as general-purpose multiclass object detection in Microsoft COCO datasets. Furthermore, it can be observed that eliminating certain strategies and techniques such as attention modules, results in degradation of performance of NextDet. Likewise, replacing the ultra-efficient backbone, CondenseNeXt, with other popular lightweight CNNs such as MobileNetv3 and ShuffleNet, degrades NextDet's object detection performance significantly, and appositely corresponds to performance metrics in Table 4. Thus, the experiments conclude that the backbone and head of an object detector play a crucial role in boosting an object detector's detection performance than the neck because backbone emphasizes on locating an object in an input image, and the head focus on detecting and refining the location of the bounding boxes.

The proposed NextDet object detector is a combination of such novel and modified techniques and strategies in the backbone, neck and head which results in a notable performance benchmarked in Tables 2 and 3. In addition, Figure 9 provides a heatmap of class activation mapping of an input image from the COCO dataset, using gradients of two objects passing through final convolution layers of the proposed object detection network highlighting salient regions of the input image crucial for the object detection task. Finally, Figure 10 provides the results of object detection on the validation set of images of Argoverse-HD and COCO datasets respectively.

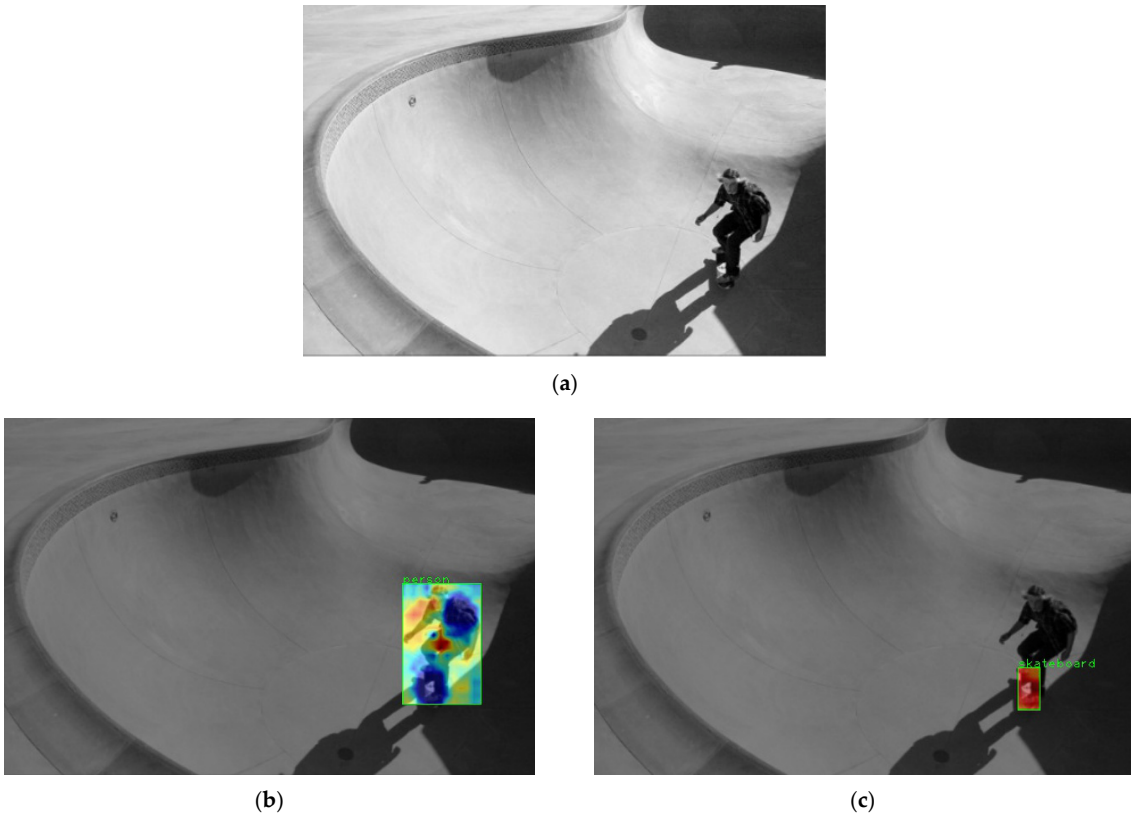

Figure 9. An example of a person riding a skateboard image from COCO dataset's training set. Image (**a**) is an input image, and images (**b**,**c**) are heatmaps of class activation mapping of two object classes: a person and a skateboard respectively. The red regions of the heatmap indicate stronger activations utilized for prediction by the proposed NextDet object detection network. Darker regions of (**b**,**c**) indicates points in the image where no activations have occurred.

(**a**) (**b**)

Figure 10. Test results of the proposed NextDet's sparse and dense object detection along with confidence scores on the validation set of images. Image (**a**) provides object detection result on the Argoverse-HD dataset, and image (**b**) provides object detection result on the COCO dataset.

5. Conclusions

Object detection is a complex computer vision task that involves image classification to predict class of an object, and object localization to identify where the object is located within an image and draw a bounding box(s) around it. The scope of the work presented within this paper proposes a modern object detection network to efficiently detect objects of multiple classes within an image using a modified version of the highly efficient CondenseNeXt CNN as the backbone along with path aggregation network connected to the feature pyramid network and spatial pyramid pooling in the neck, and spatial attention module blocks for attentive feature aggregation in the head design of the proposed object detector's architecture. Extensive analysis has been performed on two publicly available multi-class datasets: Argoverse-HD and Microsoft COCO, which provide numerous temporarily sparse to dense annotated images and bounding boxes for object detection, tracking and forecasting for autonomous driving, everyday objects, and humans. To further test robustness and obtain a fair comparison result, other PyTorch versions of lightweight CNNs are replaced and attached as backbone with NextDet's architecture and benchmarked. NextDet results in good performance and an impressive interpretability on Argoverse-HD's monocular ego-vehicle camera captured scenarios and COCO dataset's images of everyday scenes of common objects in their natural habitat.

Author Contributions: Conceptualization, P.K. and M.E.-S.; methodology, P.K.; software, P.K.; validation, P.K. and M.E.-S.; writing—original draft preparation, P.K.; writing—review and editing, P.K. and M.E.-S.; supervision, M.E.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Openly available public datasets such as Argoverse-HD and COCO have been utilized in the study. They are cited in references [9,10] of this paper.

Acknowledgments: The authors like to acknowledge the Indiana University Pervasive Technology Institute for providing supercomputing and storage resources as well as the Internet of Things (IoT) Collaboratory at the Purdue School of Engineering and Technology at Indiana University Purdue University at Indianapolis that have contributed to the research results reported within this paper.

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

References

1. Jiang, Y.; Li, X.; Luo, H.; Yin, S.; Kaynak, O. Quo Vadis Artificial Intelligence? *Discov. Artif. Intell.* **2022**, *2*, 4. [CrossRef]
2. Pang, Y.; Cao, J. Deep Learning in Object Detection. In *Deep Learning in Object Detection and Recognition*; Jiang, X., Hadid, A., Pang, Y., Granger, E., Feng, X., Eds.; Springer: Singapore, 2019; pp. 19–57. ISBN 978-981-10-5152-4.

3. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
4. Kalgaonkar, P.; El-Sharkawy, M. CondenseNeXt: An Ultra-Efficient Deep Neural Network for Embedded Systems. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Virtual Conference, 27–30 January 2021; pp. 524–528.
5. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
6. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
8. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
9. Argoverse-HD. Available online: https://www.kaggle.com/datasets/mtlics/argoversehd (accessed on 23 September 2022).
10. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2015**, arXiv:1405.0312.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325.
12. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Salt Lake City, UT, USA, 18–23 June 2018.
13. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
14. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019.
15. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-Based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1605.06409.
16. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to Compare: Relation Network for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 July 2014.
18. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [CrossRef] [PubMed]
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *arXiv* **2018**, arXiv:1703.06870.
21. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
22. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards Balanced Learning for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
23. Kim, J.; Sung, J.-Y.; Park, S. Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics—Asia (ICCE-Asia), Seoul, Korea, 1–3 November 2020; pp. 1–4.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *Proceedings of the Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 346–361.
25. Liu, S.; Huang, D.; Wang, Y. Receptive Field Block Net for Accurate and Fast Object Detection. In *Proceedings of the Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 404–419.
26. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.
27. Huang, G.; Liu, S.; Maaten, L.V.D.; Weinberger, K.Q. CondenseNet: An Efficient DenseNet Using Learned Group Convolutions. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2752–2761.
28. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3 2019. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
29. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.

30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Du, X.; Lin, T.-Y.; Jin, P.; Ghiasi, G.; Tan, M.; Cui, Y.; Le, Q.V.; Song, X. SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
32. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
33. YOLOv5 Documentation. Available online: https://docs.ultralytics.com/ (accessed on 23 September 2022).
34. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
35. Krizhevsky, A. Convolutional Deep Belief Networks on CIFAR-10. 2012; 9, *Unpublished manuscript*.
36. Alkhouly, A.A.; Mohammed, A.; Hefny, H.A. Improving the Performance of Deep Neural Networks Using Two Proposed Activation Functions. *IEEE Access* **2021**, *9*, 82249–82271. [CrossRef]
37. Zhang, Y.; Jin, R.; Zhou, Z.-H. Understanding Bag-of-Words Model: A Statistical Framework. *Int. J. Mach. Learn. & Cyber.* **2010**, *1*, 43–52. [CrossRef]
38. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19.
39. Chang, M.-F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3D Tracking and Forecasting With Rich Maps. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8740–8749.
40. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]
41. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Available online: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf (accessed on 10 October 2022).
42. Stewart, C.A.; Welch, V.; Plale, B.; Fox, G.; Pierce, M.; Sterling, T. Indiana University Pervasive Technology Institute. Available online: https://scholarworks.iu.edu/dspace/handle/2022/21675 (accessed on 10 October 2022).
43. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.