Transport Object Detection in Street View Imagery Using Decomposed Convolutional Neural Networks

Bai, Cloud; Shang, Changjing; Li, Ying; Shen, Liang; Zeng, Xianwen; Shen, Qiang

Published in:
Advances in Computational Intelligence Systems

DOI:
10.1007/978-3-031-55568-8_34

Publication date:
2024

Citation for published version (APA):
Bai, C., Shang, C., Li, Y., Shen, L., Zeng, X., & Shen, Q. (2024). Transport Object Detection in Street View Imagery Using Decomposed Convolutional Neural Networks. In *Advances in Computational Intelligence Systems* https://doi.org/10.1007/978-3-031-55568-8_34

Transport Object Detection in Street View Imagery Using Decomposed Convolutional Neural Networks

Yunpeng Bai[1][0000-0002-6923-672X] and Changjing Shang[1], Ying Li[2], Liang Shen[3], Xianwen Zeng[4], and Qiang Shen[1]

[1] |Dept. of Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, UK
[2] School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China
[3] School of Information Engineering, Fujian Business University, Fuzhou 350506, China
[4] Fujian Yingge Information Technology Co. Ltd., Fuzhou 350000, China
yub3@aber.ac.uk;cns@aber.ac.uk;lybyp@nwpu.edu.cn;liang.shen.18@fjbu.edu.cn; yingge_info@163.com;qqs@aber.ac.uk

Abstract. Deep learning has achieved great successes in performing many visual recognition tasks including object detection. Nevertheless, existing deep networks are computationally expensive and memory intensive, hindering their deployment in resource-constrained environments, such as mobile or embedded devices that are widely used by city travellers. Recently, estimating city-level travel patterns using street imagery has shown to be a potentially valid way according to a case study with Google Street View (GSV), addressing a critical challenge in transport object detection. This paper presents a compressed deep network using tensor decomposition to detect transport objects in GSV images, which is sustainable and eco-friendly. In particular, a new dataset named Transport Mode Share-Tokyo (TMS-Tokyo) is created to serve the public for transport object detection. This is based on the selection and filtering of 32,555 acquired images that involve 50,827 visible transport objects (including cars, pedestrians, buses, trucks, motors, vans, cyclists and parked bicycles) from the GSV imagery of Tokyo. Then a compressed convolutional neural network (termed SVDet) is proposed for street view object detection via tensor train decomposition on a given baseline detector. Experimental results conducted on the TMS-Tokyo dataset demonstrate that SVDet can achieve promising performance in comparison with conventional deep detection networks.

Keywords: Convolutional Neural Networks, Street-view Object Detection, Tensor Train Decomposition.

1 Introduction

Object detection is a vital branch of computer vision, aiming to locate the exact locations of target objects from complex images while determining the specific category of every object by annotating its bounding box. Particularly, the transport object detection task in street view images is to determine whether a street view image contains multiple transport objects belonging to the class of interest. Convolutional neural

network (CNN) based detectors can be technical enablers with significant potential for such applications, offering a great advantage in terms of detection accuracy over traditional pattern matching-based algorithms. However, hardware implementation of deep learning is restricted by the model size and the number of floating-point operations required. Whilst the usual millions of parameters in a convolutional model may have a powerful expression after training, the storage and loading of these parameters have high requirements on memory and disk, and meanwhile, the computation of convolutional operations on high-resolution images is often substantial. In the development of deep learning, the usual massive model size is not considered sustainable and eco-friendly in the long term due to its massive parameters and lengthy training, With the consideration of sustainability and environmental impacts, how to implement the research results based on deep learning in a cost-effective manner has emerged as an urgent challenge for the machine vision community. One of the prevailing trends is to compress the model size by a certain percentage, while still expecting the model performance to attain a relatively decent level of accuracy.

Recent studies have indicated that city transport conditions have a significant impact upon future urban planning as well as upon the public health [22]. In recognition of this point, it would be beneficial to investigate the transportation mode share in a given city, in order to assess travel patterns and transport use. Transport mode share is an essential reference for urban planning domain, serving as a strategic method for the development of Smart Cities [12]. Particularly, street imagery has proven to be a promising data source that provides visual information of the streets globally, typically in the form of panoramic images [11]. Compared with traditional methods for travel surveys, street view counts facilitate a more cost-effective approach for transport mode share analysis. Inspired by this observation, a new dataset named Transport Mode Share-Tokyo (TMS-Tokyo) is herein developed to provide a basis upon which to conduct transport object detection and city-level transport mode share analysis.This is carried out in an effort to achieve improved performance for processing street view imagery in resource-constrained environments over existing object detection approaches, with a higher accuracy and detection speed.

The main contributions of this paper are as follows. TMS-Tokyo is the largest annotated transport object dataset with aimed categories to date, offering a significant potential to develop and examine detectors designed for public road users. In particular, 32,555 images are selected and filtered that contain 50,827 visible transport objects from GSV imagery of Tokyo. The images are manually annotated individually-with bounding box annotations into 8 categories of target road users, which include cars, pedestrians, buses, trucks, motors, vans, cyclists and parked bicycles. A compressed convolutional network (SVDet) is then constructed for transport object detection based on Tensor Train (TT) decomposition. Compared with the baseline model RetinaNet [18] that represents the state-of-the-art in the relevant literature, SVDet achieves a mAP gain of 0.9%, while saving more than 68.8% of parameters and 52.3% computational time.

The rest of this paper is organised as below. Section 2 presents a brief review of the relevant background. Section 3 introduces in detail of the dataset. Section 4 describes the proposed approach. Section 5 provides an experimental study and discuss-

es the results in comparison with the existing literature. Finally, Section concludes this research and points out interesting further work.

2 Related Work

For academic completeness, the state-of-the-art deep learning based techniques for object detection in general, and the specific approach for compression-based model decomposition in particular are herein introduced.

2.1 Object Detection Models

A modern CNN-based object detector is usually composed of three consecutive parts, a backbone, a neck and a head. The backbone that is used for image feature extraction may often be implemented via VGG [29], ResNet [13], DenseNet [14]. The neck is devised to exploit the features extracted from different stages by the backbone, normally consisting of several bottom-up paths and several top-down paths. Typical neck modules may include Feature Pyramid Network (FPN) [17], Path Aggregation Network (PAN) [19], BiFPN [31], and NAS-FPN [9]. The head, which is used to predict classes and bounding boxes of objects, is usually categorised into two types, namely, one-stage detector and two-stage detector. The most representative two-stage detectors are those belonging to the R-CNN [10] series (including fast R-CNN [18], faster R-CNN [26], R-FCN [5], and Libra R-CNN [24]), and the most one-stage representative models are YOLO [2, 25], SSD [20], and RetinaNet [18].

2.2 Low-rank Decomposition Based Model Compression

Model compression and acceleration refer to the distillation of redundant parameters in a neural network to obtain a small-scale model with fewer parameters and a more compact structure, under a certain degree of algorithm completion. Low-rank filters have been utilised to accelerate convolution for a long history (e.g., separable 1D filters were introduced using a dictionary learning approach [27]). Regarding deep neural network (DNN) models, efforts have also been made for low-rank approximation, as reported in [7]. In such work, the speed of a single convolutional layer was increased by a factor of 2, but the classification accuracy was decreased by 1%. In [15], a different tensor decomposition scheme was proposed, achieving a 4.5-fold speedup with the same rate of accuracy loss.

There exist a number of low-rank methods for compressing 3D convolutional layers. For instance, Canonical Polyadic (CP) decompositions for kernel decomposition adopt nonlinear least squares to compute the CP decomposition [16]. Also, batch normalization (BN) is employed to transform the activation of the internal hidden units [30], aiming at training low-rank constrained CNNs from scratch. Meanwhile, many approaches have been proposed to exploit low-rankness in the fully connected layers, including the use of low-rank methods to reduce the volume of dynamic parameters [6]. A specific development is for acoustic modelling, where a low-rank

matrix factorization of the final weight layer is introduced in the DNN [28]. In order to compact deep learning models for multi-tasks, the truncated singular value decomposition (SVD) has been well adapted to decompose fully connected layers in order to develop compact multitask deep learning architectures [21]. Of direct interest to the present work is the attempt to adopt TT decomposition to compress the convolutional layers and fully connected layers in a network, which entails significant compression rates only with a slight drop in accuracy.

3 TMS-Tokyo Dataset

Object detection is a vital task to be addressed in computer vision while large datasets for training is instrumental in developing the relevant techniques in the subject area. There are a variety of street-view datasets in computer vision. Indeed, many open-source databases are available that reflect diversity and richness in terms of category types and sample sizes and thereby, may be utilised to support performing a range of machine vision tasks. However, urban mobility needs cannot be satisfied with such datasets mainly due to their ineligible sample categories captured with different ratios. To aid in advancing transport object detection research in street view scenes, this section introduces a large-scale street-view imagery dataset named Transport Mode Share-Tokyo Dataset (TMS-Tokyo).

Google Street View Imagery eliminates the difficulty of capturing a high-resolution perspective view of scenes with rich color and texture information, enabling the gathering of accurate, timely and representative mobility data, which is a trivial task for machine vision[1]. In preparation of the TMS-Tokyo dataset, the bounding boxes of 32,555 Google Street View images are herein manually annotated involving a total of 50,827 labelled transport objects of eight categories. Each GSV image is of a fixed size of 512×512 pixels, containing transport objects in different scales on the road.

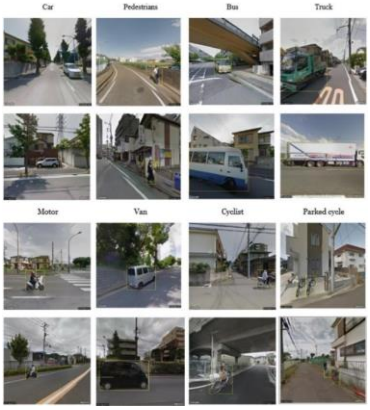

Fig. 1. Illustrative transport mode samples with annotations

This dataset consists of a massive amount of eight defined classes of transport objects that vary widely in appearance, scale, occlusion and viewpoint. In particular, the target categories of road users include cars, pedestrians, buses, trucks, motor, vans, cyclists and parked cycles. This is implemented by reviewing the common transport modes that frequently appear in Tokyo. Notably, single instances of road users are annotated individually. An excerpt of eight transport mode samples is illustrated in Fig. 1. To reflect the complexity of the introduced dataset, the basic properties of TMS-Tokyo are listed in Table 1, together with those of other more established transport datasets, including Tsinghua-Daimler Cyclist Benchmark [33], KITTI [8], and CityScape [4]. Note that whilst TMS-Tokyo is not the largest image dataset introduced for object detection, it is the Google Street View dataset of the larges size specifically devised for transport object detection.

Table 1. Street-view image datasets involving transport objects, where BBox stands for bounding box.

Dataset	Annotation	#Categories	#Instances	#Images
Tsinghua-Daimler Cyclist [33]	Horizontal BBox	7	32,361	14,674
Cityscape [4]	Segmentation	30	NA	25,000
KITTI [8]	Segmenation	5	80,256	14,999
Mapillary Vistas [23]	Segmentation	66	>2 M	25,000
BDD 100K [34]	Horizontal BBox	10	3.3M	100,000
TMS-Tokyo	Horizontal BBox	8	50,827	33,461

4 Proposed Approach

This section presents a novel approach for the development of decomposed CNN-based transport object detection in street view imagery.

4.1 Data Augmentation

For street view imagery, the backgrounds of street views can be various and complicated in different locations and scenes. Road users often appear in different orientations, positions, scaling, and brightness. In practice, it is easy to fall into the trap of overfitting with limited data in the face of trillions of parameters in a deep neural network. The data augmentation technique helps increase the relevance of the data, thereby minimizing the possibility of neural networks learning irrelevant features, radically improving overall performance.

An interesting approach for data augmentation is the Mosaic method first proposed in YOLOv4 [2], of which the main idea is to randomly crop a small number (four in a typical implementation) of images and to stitch them onto one image as training data. This paper adopts Mosaic data augmentation technique on the dataset TMS-Tokyo, while turning off this operation in a number (say, 15) of the last epochs of training to prevent the images generated by data enhancement from interfering with the real distribution of natural images. An instance of Mosaic in TMS-Tokyo is illustrated in Fig. 2.

Fig. 2. Examples of Mosaic data augmentation.

4.2 Tensor Train Decomposition

Tensor Train (TT) decomposition is a tensor chain decomposition based on Matrix Product State (MPS) model, which decomposes input tensor into a series of adjacent three-dimensional and two-dimensional tensors. Typically, TT decomposition can be achieved by (*N-1*) times singular value decomposition. For a fourth-order tensor, for instance, the decomposition takes the following form:

$$X(i,j,k,l)=\sum_{r1,r2,r3,r4} G1(i,r_1)G2(r_1,j,r_2)G3(r_2,k,r_3)G4(r_3,l) \tag{1}$$

The process of such an algorithm can be divided into four steps as follows:

Step 1. Decomposing the convolutional kernel in the original neural network into a fourth-order core tensor, producing a factor matrix by TT decomposition.

Step 2. Completing the factor matrix of the decomposition and filling the convolutional kernel parameters.

Step 3. Assigning the new convolutional kernel parameters to the new convolutional kernel.

Step 4. Replacing the original convolution kernel with the new two-layer mini-convolution.

The advantage of TT decomposition is that it is linear in relation to the number of entries (and hence, storage) and computation time, enabling higher dimensional problems to be addressed. In particular, when dealing with matrices, the TT decomposition is equivalent to the singular value decomposition..

Finding the optimal rank is a key issue when compressing the model through low rank decomposition. Rank is the only hyperparameter that controls computational complexity and accuracy in compressed convolutional neural networks. An excessively large rank clearly does not achieve maximum compression, whilst a rank that is too small may make accuracy recovery problematic. Instead of choosing the rank by time-consuming iterative trials, the Empirical Variational Bayes Matrix Factorization (EVBMF) method is employed to automatically compute the rank.

In this work, a superior selection is shown to be attainable for full Variational Bayes Matrix Factorization (VBMF). More specifically, the global optimum is a re-weighted SVD of the observation matrix, and each weight can be obtained by solving a quadratic equation whose coefficients are a function of the observed singular values. Therefore, EVBMF, where the hyperparameters learned from the data, is adopted in our work to achieve the global optimal solution.

4.3 Overall Structure of SVDet

Recall the original design intention, which is to develop a compact and high-precision model for traffic object detection. Considering the objective of dealing with (and hence, combining the metrics of) both detection accuracy and model complexity, the RetinaNet model is chosen as the baseline detector to perform low-rank decomposition in an effort to obtain a compact model. In implementation, SVDet consists of two steps: first decomposing and replacing the convolutional kernels (network weights) of the backbone and head parts of RetinaNet using the TT decomposition algorithm, and then fine-tuning the compressed model to reduce the impact of the decomposition on the resulting model accuracy.

Table 2 presents the statistics of computational and parametric quantities for the backbone, neck and head parts of RetinaNet with Mosaic data augmentation for the dataset investigated. It can be seen that the backbone, as the feature extraction part of the detection model, accounts for a relatively large amount of computation and number of parameters, and the number of parameters accounts for 64.2% of the entire model. The head part is detected on multiple feature layers and its parameters are shared, so that the number of parameters in it is low but the computation is larger, reaching 51.1% of the entire model. In this paper, low-rank decomposition is performed on these two components. The overall algorithm structure of the low-rank decomposition of the backbone network of SVDet using TT decomposition is shown in Fig. 4.

Table 2. FLOPs and Params for different parts of baseline model

Parts	FLOPs (G)	Params (M)
Baseline	53.07	36.25
Backbone	21.52(40.6%)	23.28(64.2%)
Neck	4.42(8.3%)	8.0(22%)
Head	27.13(51.1%)	4.97(13.7%)

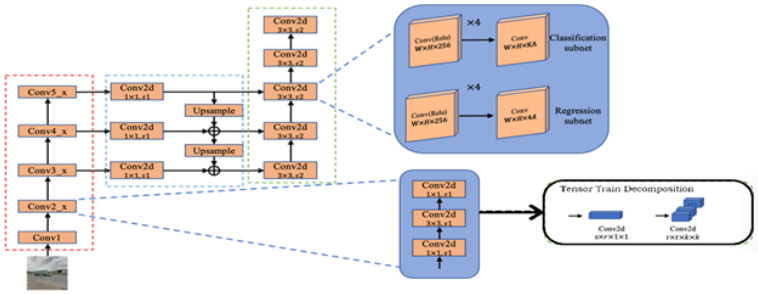

Fig. 3. Algorithm structure of SVDet, where red, blue and green dotted boxes indicate Backbone, Neck, and Head part of SVDet, respectively

5 Experimental Results and Analysis

The performance of DVNet is examined in comparison with other baselines on the TMS-Tokyo Dataset in accomplishing the multi-class transport object detection task. The algorithm benchmarks employed are the state-of-the-art deep CNN methods with proven superior performance and representativeness when applied for object detection, including ResNet-50 and ResNet-101 based RetinaNet [18], Faster RCNN [26], Cascade RCNN [3], FCOS [32], and Darknet-based YOLOv3 [25].

5.1 Implementation configuration

All experiments reported in the paper were performed on a 64-bit Ubuntu 18.04 operating system. The hardware is a GTX 1080 Ti GPU with 12GB of RAM and a 10-core Intel(R) Xeon(R) CPUE5-2640 v4@2.40GHz CPU with 32GB of RAM. The entire dataset is divided into three sets of training, validation and testing according to the ratio of 6:2:2. In presenting the following experimental results, the unit of FLOPs is GFLPOs (1GFLOPs = 109FLOPs) and that of Params is megabytes (M).

5.2 Comparative Results

The results of SVDet are compared with those achieved by the other object detectors investigated on the TMS-Tokyo Dataset, as listed in Table 3. Judged by mean Average Precision (mAP), SVDet outperforms the others. Through its balanced design, SVDet achieves 77.6 on mAP with ResNet-50. In particular, as a lightweight single-stage object detection method, SVDet significantly outperforms both typical one-stage and two-stage methods in terms of computational parameters. From the perspective of overall efficiency and effectiveness, SVDet is shown to be the most promising transport object detector.

Table 3. Numerical results based on TMS-Tokyo

Methods	Backbone	mAP (%)	FLOPs (G)	Params (M)
Faster R-CNN [12]	R50	76.20	63.28	41.16
	R101	76.60	82.76	60.15
Cascade R-CNN [45]	R50	76.80	91.07	69.95
	R101	77.10	110.55	87.94
YOLOv3 [16]	Darknet	76.10	49.66	61.56
FCOS [46]	R50	74.50	50.41	31.85
	R101	75.60	69.88	50.79
RetinaNet [18]	R50	75.90	53.07	36.25
	R101	76.30	72.55	55.24
ATSS [47]	R50	75.30	51.63	31.90
SVDet	R50	**77.60**	**16.52**	**17.29**

5.3 Ablation Studies

For ablation experiments, the SVDet model trained on TMS-Tokyo serves as the pre-trained model for network initialization. In so doing, all model parameters are distributed in a performance-strong range at the beginning of training, alleviating potential overfitting while speeding up the convergence of the underlying model. RetinaNet is taken as the baseline due to its promising performance among classical detectors. To validate the design of the proposed model, experiments are conducted to test the influence of different model compression methods and that of low-rank decomposition of different modules, regarding the model performance and the number of computational parameters respectively.

5.3.1 Comparison with other model compression algorithms

A comparative experimental test is carried out against the baseline algorithm and three other classical model compression algorithms, namely CP decomposition, Tucker decomposition and Stripe-Wise Pruning.

Table 4. Comparison with other model compression algorithms.

Methods	mAP (%)	FLOPs (G)	Params (M)
Baseline	76.9	53.07	36.25
Stripe-Wise Pruning	72.37(5.9% ↓)	31.31(41.0% ↓)	18.85(48.0% ↓)
CP decomposition	56.80(26.1% ↓)	**11.42(78.5% ↓)**	**12.72(64.9% ↓)**
Tucker decomposition	68.20(11.3% ↓)	15.21(71.3% ↓)	14.68(59.5% ↓)
TT decomposition	**77.60(0.9% ↑)**	16.52(68.9% ↓)	17.29(52.3% ↓)

The results as given in Table 4 show that while the other three methods experienced accuracy loss SVDet (i.e., TT decomposition) gained higher accuracy instead. Particularly, in terms of computational and parametric quantities, CP decomposition reduced the maximum complexity but its accuracy significantly fallen behind the rest. In summary, SVDet is demonstrated to be a low-rank decomposition that can balance the accuracy loss and computational parametric reduction, beating the existing model compression algorithms.

5.3.2.Influence of decompositions on different model modules

To investigate the influence of decompositions of different modules on model performance and computational capacity, a number of experiments are devised to decompose different components of the baseline algorithm using three low-rank decompositions. Each is initialized with pre-trained model parameters, while noting that SVDet simultaneously performs a low-rank TT decomposition of both Backbone and Head modules for the baseline.

From the results of Table 5, it can be seen that the TT decomposition of Backbone and Head of the baseline method offers the most promising outcome. It achieves less

than one-half of the number of parameters and that of computation of the uncompressed model while the accuracy also gains a little over the uncompressed model.

Table 5. TT decomposition on different modules

Backbone	Head	mAP (%)	FLOPs (G)	Params (M)
×	×	76.9	53.07	36.25
√	×	**78.10(1.6% ↑)**	41.43(21.9% ↓)	21.85(39.7% ↓)
×	√	77.80(1.2% ↑)	28.17(46.9% ↓)	31.69(12.6% ↓)
√	√	77.60(0.9% ↑)	**16.52(68.9% ↓)**	**17.29(52.3% ↓)**

Amongst the three components implemented for object detection, the maximum gain is achieved when performing low-rank decomposition on both the Backbone and the Head, because these components own the majority of the entire model parameters. Tucker decomposition is better than CP decomposition, but both of them have limitations: the former requires more storage space while lacking correlation information between any two patterns, and the latter lacks correlation information between the tensor and other different patterns that Tucker decomposition can obtain. However, the tensor train (TT) decomposition shows a strong higher-order processing capability and is well suited for the decomposition of neural network models. This is feasible because SVDet is of a fourth-order tensor, with TT having a great advantage for the third-order and above.

6 Conclusion

This paper has proposed an innovative approach using deep learning to detect transport objects from Google Street View imagery. A GSV imagery dataset (named TMS-Tokyo) has been introduced for the first time, involving eight categories of road users and containing 50,827 instances and 32,555 images. It is the largest Google Street View dataset specially designed for transport object detection. To adopt and reflect evolving urban transport conditions, TMS-Tokyo will be continuously updated and extended in size and scope, by involving more cities on a global scope.

Further to the introduction of a new dataset, low-rank tensor decomposition has been proposed to compress the street view object detector from the perspective of compression parameters, in order to tackle the challenging problem of high computational cost and parametric volume of the detection model. The resulting system SVDet applies Tensor Train decomposition to both the Backbone and the Head of the underlying model. It is able to achieve a mAP value of 77.6% on TMS-Tokyo, with a parametric number of 17.29M and a computational volume of 16.52GFLOPs, significantly outperforming the state-of-the-art methods in the literature. Thanks to such a lightweight design based on low-rank tensor decomposition the present approach can contribute to helping address the important issue of environmental sustainability. How this system may be further developed to cope with the TMS-Tokyo dataset that is to be significantly expanded remains active research.

Acknowledgements

This work is supported in part by the Strategic Partner Acceleration Award (80761-AU201), funded under the Ser Cymru II programme, UK. The first author is supported with a full International PhD Scholarship awarded by Aberystwyth University.

References

1. Anguelov, D. et al.: Google Street View: Capturing the World at Street Level. Computer. 43, 6, 32–38 (2010). https://doi.org/10.1109/MC.2010.170.
2. Bochkovskiy, A. et al.: YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs, eess]. (2020).
3. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving into High Quality Object Detection. arXiv:1712.00726 [cs]. (2017).
4. Cordts, M. et al.: The Cityscapes Dataset for Semantic Urban Scene Understanding. arXiv:1604.01685 [cs]. (2016).
5. Dai, J. et al.: R-FCN: Object Detection via Region-based Fully Convolutional Networks. arXiv (2016). https://doi.org/10.48550/arXiv.1605.06409.
6. Denil, M. et al.: Predicting Parameters in Deep Learning. arXiv:1306.0543 [cs, stat]. (2014).
7. Denton, E. et al.: Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation. arXiv:1404.0736 [cs]. (2014).
8. Geiger, A. et al.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074.
9. Ghiasi, G. et al.: NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. arXiv:1904.07392 [cs]. (2019).
10. Girshick, R. et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 [cs]. (2014).
11. Goel, R. et al.: Estimating city-level travel patterns using street imagery: A case study of using Google Street View in Britain. PLOS ONE. 13, 5, e0196521 (2018). https://doi.org/10.1371/journal.pone.0196521.
12. Grimsrud, M., El-Geneidy, A.: Transit to eternal youth: lifecycle and generational trends in Greater Montreal public transport mode share. Transportation. 41, 1, 1–19 (2014). https://doi.org/10.1007/s11116-013-9454-9.
13. He, K. et al.: Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs]. (2015).
14. Huang, G. et al.: Densely Connected Convolutional Networks. arXiv (2018). https://doi.org/10.48550/arXiv.1608.06993.
15. Jaderberg, M. et al.: Speeding up Convolutional Neural Networks with Low Rank Expansions. arXiv:1405.3866 [cs]. (2014).
16. Lebedev, V. et al.: Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition. arXiv:1412.6553 [cs]. (2015).

17.Lin, T.-Y. et al.: Feature Pyramid Networks for Object Detection. arXiv:1612.03144 [cs]. (2017).

18.Lin, T.-Y. et al.: Focal Loss for Dense Object Detection. arXiv:1708.02002 [cs]. (2018).

19.Liu, S. et al.: Path Aggregation Network for Instance Segmentation. arXiv (2018). https://doi.org/10.48550/arXiv.1803.01534.

20.Liu, W. et al.: SSD: Single Shot MultiBox Detector. arXiv:1512.02325 [cs]. 9905, 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2.

21.Lu, Y. et al.: Fully-Adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1131–1140 (2017). https://doi.org/10.1109/CVPR.2017.126.

22.Mueller, N. et al.: Health impacts related to urban and transport planning: A burden of disease assessment. Environment International. 107, 243–257 (2017). https://doi.org/10.1016/j.envint.2017.07.020.

23.Neuhold, G. et al.: The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5000–5009 (2017). https://doi.org/10.1109/ICCV.2017.534.

24.Pang, J. et al.: Libra R-CNN: Towards Balanced Learning for Object Detection. arXiv:1904.02701 [cs]. (2019).

25.Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs]. (2018).

26.Ren, S. et al.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497 [cs]. (2016).

27.Rigamonti, R. et al.: Learning Separable Filters. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2754–2761 (2013). https://doi.org/10.1109/CVPR.2013.355.

28.Sainath, T.N. et al.: Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 6655–6659 (2013). https://doi.org/10.1109/ICASSP.2013.6638949.

29.Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv (2015). https://doi.org/10.48550/arXiv.1409.1556.

30.Tai, C. et al.: Convolutional neural networks with low-rank regularization. arXiv:1511.06067 [cs, stat]. (2016).

31.Tan, M. et al.: EfficientDet: Scalable and Efficient Object Detection. arXiv (2020). https://doi.org/10.48550/arXiv.1911.09070.

32.Tian, Z. et al.: FCOS: Fully Convolutional One-Stage Object Detection. Presented at the (2019).

33.Xiaofei Li et al.: A new benchmark for vision-based cyclist detection. In: 2016 IEEE Intelligent Vehicles Symposium (IV). pp. 1028–1033 (2016). https://doi.org/10.1109/IVS.2016.7535515.

34.Yu, F. et al.: BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. arXiv:1805.04687 [cs]. (2020).