Pedestrian in Traffic Dataset

Swaroop A. Naik

12101735

RQ1E35B20

Abstract

Pedestrian accidents most frequently occur in complex circumstances, because of many factors related to the behavior of drivers and pedestrians. The basic parameters that determine road safety
include the perception of traffic and visibility on the road. At present, pedestrian crossings are one of the greatest challenges for traffic and safety engineering. The pedestrian mode is an important component of urban networks, and greatly affects the performance of the sidewalks and crosswalks, as well as the entire network traffic operations by interacting with other traffic modes (automobile, bicycle, transit). There have been many studies concerning different aspects of pedestrian behaviors, such as pedestrian walking speed, pedestrian delay, gap acceptance, signal compliance, route choice, etc. A high-risk area for pedestrians, the unsignalized midblock crosswalk is a conflict zone between pedestrians and vehicles. In this paper, the behaviors and traffic characteristics of pedestrians walking through the unsignalized midblock crosswalk without being disturbed by other pedestrians are analyzed. The data, including pedestrian speed, waiting for the delay, and clustering, are obtained by counting and measuring with a video camera. Pedestrian behaviors are analyzed with the use of comparisons between various categories and through statistical analysis. Pedestrian behaviors are interpreted further by analyzing pedestrians' tactics when they cross the street and the valid gap in the vehicle flow, based on observations and measurements of pedestrians starting to walk across and of vehicle arrivals.

1. INTRODUCTION

Pedestrian detection is used in many vision-based applications ranging from video surveillance to autonomous driving. Despite achieving high performance, it is still largely unknown how well existing detectors generalize to unseen data. This is important because a practical detects- tor should be ready to use in various scenarios in applications. To this end, we conduct a comprehensive study in this paper, using a general principle of direct cross-dataset evaluation. The existing state-of-the-art pedestrian detectors, though perform quite well when trained and tested on the same dataset, generalize poorly in cross-dataset evaluation. We demonstrate that there are two reasons for this trend. Firstly, their designs (e.g., anchor settings) may be biased towards popular benchmarks in the traditional single-dataset training and test pipeline, but as a result, largely limit their generalization capability. Secondly, the training source is generally not dense in pedestrians and diverse in scenarios. Under direct cross-dataset evaluation, surprisingly, we find that a general-purpose object detector, without pedestrian-tailored adaptation in design, generalizes much better compared to existing state-of-the-art pedestrian detectors. Furthermore, we illustrate those diverse and dense datasets, collected by crawling the web, serve to be an efficient source of pre-training for pedestrian detection. Accordingly, we pro- pose a progressive training pipeline and find that it works well for autonomous-driving-oriented pedestrian detection.

The pedestrian mode is an important component of urban networks, and greatly affects the performance of the sidewalks and crosswalks, as well as the entire network traffic operations by interacting with other traffic modes (automobile, bicycle, transit). A schematic of a pedestrian trip in an urban network has been assumed. The trip consists of walking portions that do not have interactions with vehicles and crossing portions that do. Given an origin-destination, pedestrians have multiple route alternatives and may encounter different traffic conditions along their path. Pedestrian trip travel time represents the total time a pedestrian spends from an origin to a destination within a network. There have been many studies concerning different aspects of pedestrian behaviors, such as pedestrian walking speed, pedestrian delay, gap acceptance, signal compliance, route choice, etc.

Pedestrian traffic is a natural part of non-motorized transport, and its frequency is particularly noticeable in built-up areas in cities, villages, or places where human efforts are concentrated. It is in such built-up areas where short-distance trips mostly occur. Along with another type of non-motorized transport, which is cycling, pedestrian traffic is one of the ecological modes of transport. Compared to motor transport, they are characterized mainly by smaller space requirements and by not harming the environment.

The paper is organized as follows

- Literature Review
- Proposed Methodology
- Dataset
- Data Pre-processing
- Feature Selection
- Algorithm Used
- Results Analysis
- Reference

2. LITERATURE REVIEW

Pedestrian detection is one of the longest-standing problems in computer vision. Numerous real-world applications, such as autonomous driving [9, 17], video surveillance [16], action recognition [45], and tracking [21] rely on accurate pedestrian/person detection. Recently, convolutional neural network (CNN) based approaches have shown considerable progress in the field of pedestrian de- taction, where on certain benchmarks, the progress is within striking distance of a human baseline.

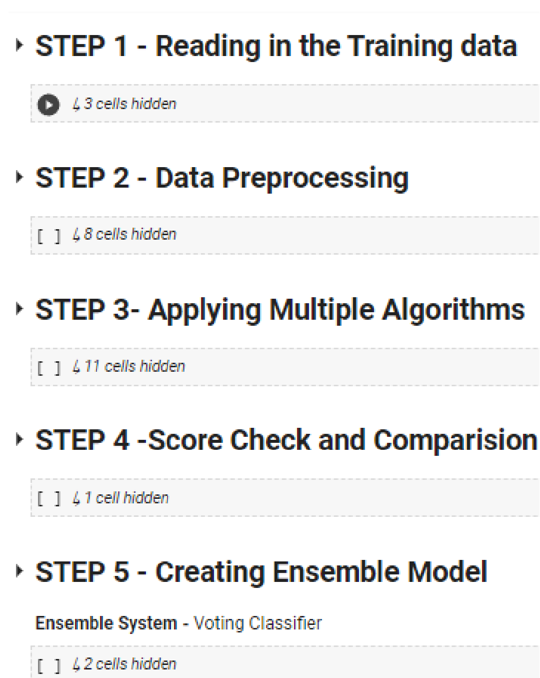
However, some current pedestrian detection methods show signs of over-fitting to source datasets, especially in the case of autonomous driving. As shown in Fig. 1 right, current pedestrian detectors, do not generalize well to other (target) pedestrian detection datasets, even when trained on a relatively large-scale dataset that is reasonably closer to the target domain. This problem prevents pedestrian detection from scaling up to real-world applications.

Despite being a key problem, generalizable pedestrian detection has not received much attention in the past. More importantly, the reasons behind the poor performances of pedestrian detectors in cross-dataset evaluation have not been properly investigated or discussed. In this paper, we argue that this is mainly because the current state-of-the-art pedestrian detectors are tailored for target datasets and their overall design is biased towards target datasets, thus reducing their generalization. Secondly, the training source is generally not dense in pedestrians and diverse in scenarios. Since current state-of-the-art methods are based on deep learning, their performance depends heavily on the quantity and quality of data and there is some evidence that the per- performance of some computer vision tasks (e.g., image classification) keeps improving by at least up to billions of samples [28].

At present, all autonomous driving-related datasets have at least three main limitations, 1) the limited number of unique pedestrians, 2) low pedestrian density, i.e., the challenging occlusion samples are relatively rare, and 3) limited diversity as the datasets are captured by a small team primarily for dataset creation instead of curating them from more diverse sources (e.g., YouTube, Facebook, etc.).

In the last couple of years, a few large and diverse datasets, Crowd Human [34], Wider Person [48], and Wider Pedestrian[1], have been collected by crawling the web and through surveillance cameras. These datasets address the above-mentioned limitations but as they are from a much broader domain, they do not sufficiently cover autonomous driving scenarios. Nevertheless, they can still be very valuable for learning a more general and robust model of pedestrians. As these datasets contain more persons per image, they are likely to contain more human poses, appearances, and occlusion scenarios, which is beneficial for autonomous driving scenarios, provided current pedestrian detectors have the innate ability to digest large-scale data.

3. PROPOSED METHODOLOGY



4. DATASET

	Time	Convention	Campus	Workplace	College	Bus Stop
0	01-01-2019 00:00	530	15	13	4	2
1	02-01-2019 00:00	639	62	21	3	8
2	03-01-2019 00:00	698	63	31	6	6
3	04-01-2019 00:00	519	47	24	6	13
4	05-01-2019 00:00	1143	52	37	19	9
...	...	...	...	...	...	...
360	27-12-2019 00:00	697	23	16	7	8
361	28-12-2019 00:00	715	16	10	4	6
362	29-12-2019 00:00	367	9	5	1	6
363	30-12-2019 00:00	488	23	12	6	8
364	31-12-2019 00:00	452	14	5	1	3

365 rows × 6 columns

Fig 4.1 Data of Pedestrians

The Dataset is self-collected and maintained. Term Explanation of Table:

- *Time* – Pertains the time slots of the pedestrian

Rest all the variables associated namely, *Convention, Campus, Workplace, College, and Bus Stop* were denoting the head counts of different pedestrians at that particulartime frame.

5. DATA PREPROCESSING

Data was first explored and read manually. The data is thoroughly analyzed for preprocessing. Columns are separated according to their dependency factor on the output needed to predict. After that, all independent variables/columns (Independent variables *(also referred to as Features)* are the input for a process that is being analyzed) are detached from the dependent variable.

Dependent variables are the output of the process. For example, in the below data set, the independent variables are the input of the purchasing process being analyzed. The result *(whether a user purchased or not)* is the dependent variable.

In the Data Cleaning process, a few targeted columns were chosen for the removal of NULL and unused values in the dataset to further prevent incorrect training of models after splitting. Data is separated in a 7:3 ratio for training and testing. Where 70% went for training and 30% for testing. Label Encoding takes place for converting the labels into a numeric form to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

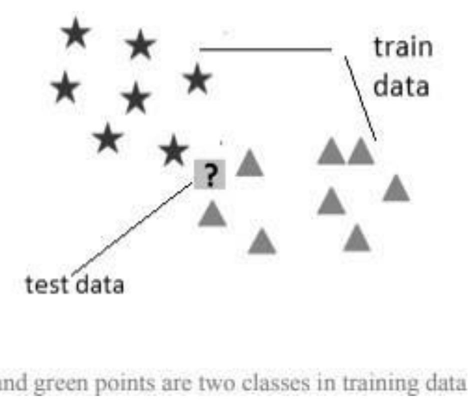


Fig 5.1 Data Split

6. FEATURE SELECTION

Independent Variable –

- Convention
- Campus
- Workplace
- College
- Bus Stop

Dependent Variable –

- Time

7. ALGORITHM USED

Prediction models are generated using four machine learning algorithms namely –

- Logistic Regression
- k-Nearest Neighbours
- Random Forest Algorithm

The attributes used in the test and train dataset for implementing these algorithms are Time, Convention,Workplace, College, and Bus Stop.

REGRESSION

Regression is a statistical method used in finance, investing, and other disciplines that attempt to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).

Regression analysis is a set of statistical methods used for the estimation of relationships between a dependent variable and one or more independent variables. It can be utilized to assess the strength of the relationship between variables and for modeling the future relationship between them. Regression analysis includes several variations, such as linear, multiple linear, and nonlinear. The most common models are simple linear and multiple linear. Nonlinear regression analysis is commonly used for more complicated data sets in which the dependent and independent variables show a nonlinear relationship.

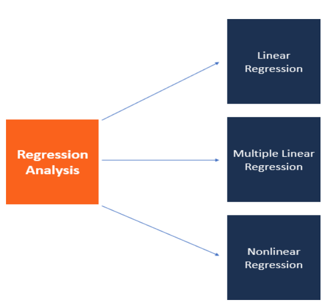


Fig 7.1 Types of Regression

Logistic Regression was used in the biological sciences in the early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable (target) is categorical. For example,

- To predict whether an email is a spam (1) or(0)
- Whether the tumor is malignant (1) or not (0)

$$Logistic\ function = \frac{1}{1+e^{-x}}$$

Fig 7.2 Sigmoid/Logistic Function

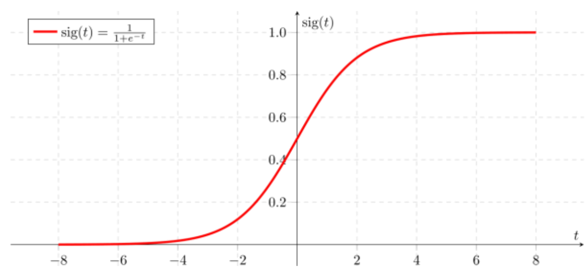


Fig 7.3 Sigmoid/Logistic Graphical

The most common logistic regression model a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

In a binary logistic regression model, the dependent variable has two levels (categorical). Outputs with more than two values are modeled by multinomial logistic regression and, if the multiple categories are ordered, by ordinal logistic regression (for example the proportional odds ordinal logistic model). The logistic regression model itself simply models the probability of output in terms of input and does not perform statistical classification (it is not a classifier), LR is a transformation of linear regression using the sigmoid function. The vertical axis stands for the probability of a given classification and the horizontal axis is the value of *x*. It assumes that the distribution of y|x is the Bernoulli distribution. The formula of LR is as follows:

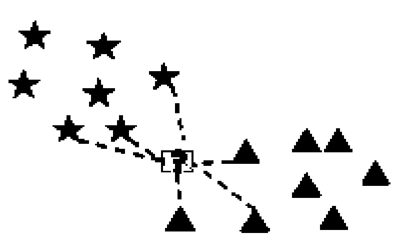
$$F(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

K-NEAREST NEIGHBORS

K-nearest neighbors (KNN) is a type of supervised learning algorithm which is used for both regression and classification purposes, but mostly it is used for classification problems. Given a dataset with different classes, KNN tries to predict the correct class of test data by calculating the distance between the test data and all the training points. It then selects the k points which are closest to the test data. Once the points are selected, the algorithm calculates the probability (in the case of classification) of the test point belonging to the classes of the k training points, and the class with the highest probability is selected. In the case of a regression problem, the predicted value is the mean of the k selected training points

Let's understand this with an illustration of the classification problem:

1) Given a training dataset as given below. We have new test data that we need to assign to one of the two classes.

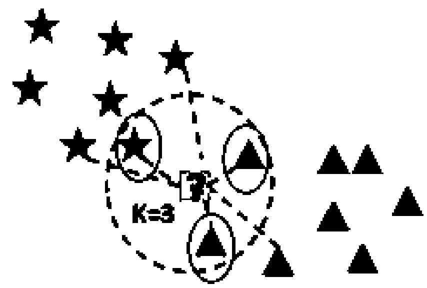


Calculating the distance between neighbor points

. *Fig 7.4 KNN*

2) Now, the k-NN algorithm calculates the distance between the test data and the given training data.

3) After calculating the distance, it will select the k training points which are nearest to the test data. Let's assume the value of k is 3 for our example



selecting k=3 where 1 red ,2 green

Fig 7.5 KNN

4) Now, 3 nearest neighbors are selected, as shown in the figure above. Let's see in which class our test data will be assigned: The number of Green class values
= 2, Number of Red class values = 1 Probability (Green) = 2/3, Probability (Red) = ⅓ Since the probability for the Green class is higher than Red, the k-NN algorithm will assign the test data to the Green class.

DECISION TREE

A Decision Tree has many analogies in real life and turns out, it has influenced a wide area of Machine Learning, covering both Classification and Regression. Sometimes Decision trees are also referred to as CART, which is short for Classification and Regression Tree. In Decision analysis, a decision tree can be used to represent decisions and decision-making visually and explicitly

Tree-Based Algorithms are a popular family of related non-parametric and Supervised methods for both classification and regression. If you're wondering what supervised learning is, it's the type of machine learning algorithm that involves training models with data that has both input and output labels. The Decision trees look like a vague upside-down tree with a decision rule at the root, from which subsequent decision rules spread out below.

There can also be nodes without any decision rules; these are called Leaf nodes. Let's see what a decision tree looks like, and how they work when a new input is given for prediction. Below is an image explaining the basic structure of the decision tree. Every tree has a root node, where the inputs are passed through. This root node is further divided into sets of decision nodes where results and observations are conditionally based. The process of dividing a single node into multiple nodes is called Splitting. If a node doesn't split into further nodes, then it's called a leaf node, or terminal node. A subsection of a decision tree is called a Branch or sub-tree.

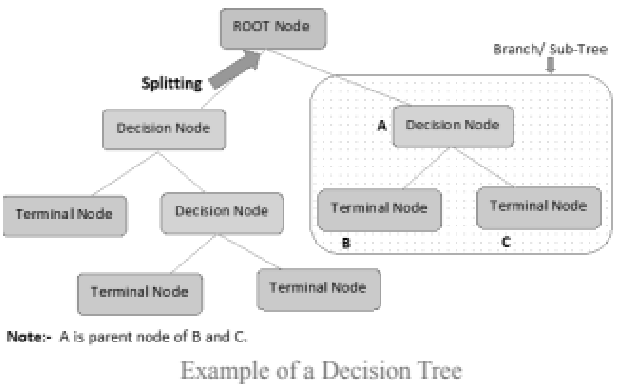


Fig 7.6 Decision Tree Working

There is also another concept that is quite opposite of splitting. If there are ever decision rules which can be eliminated, we cut them from the tree. This process is known as Pruning and is useful to minimize the complexity of the algorithm.

RANDOM FOREST

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. Ensemble uses two types of methods:

1. *Bagging*– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

2. *Boosting*– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, boost.

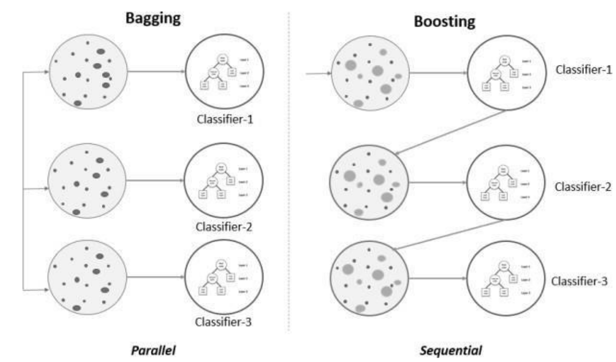


Fig 7.7 Bagging and Boosting

As mentioned earlier, Random Forest works on the Bagging principle. Now let's dive in and understand bagging in detail. Bagging, also known as Bootstrap Aggregation, is the ensemble technique used by random forests. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with a replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results.

The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as *aggregation.*

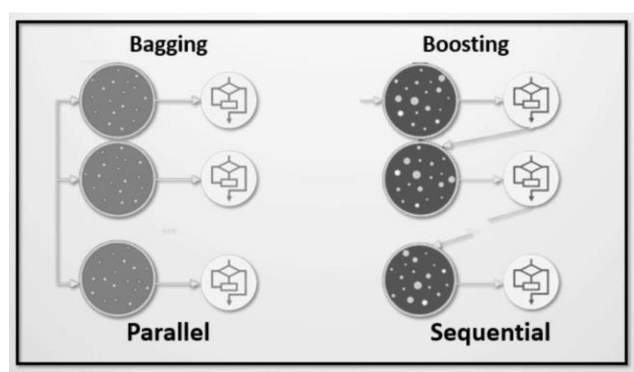


Fig 7.8 Bagging and Boosting Training

Now let's look at an example by breaking it down with the help of the following figure. Here the bootstrap sample is taken from actual data (Bootstrap sample 01, Bootstrap sample 02, and Bootstrap sample 03) with a replacement which means there is a high possibility that each sample won't contain unique data.

Now the model (Model 01, Model 02, and Model 03) obtained from this bootstrap sample is trained independently. Each model generates results as shown. Now Happy emoji is having a majority when compared to the sad emoji. Thus based on majority voting final output is obtained as Happy emoji.

8. RESULT FROM ANALYSIS

3 Algorithms were first trained and tested individually and all algorithms achieved the best scores:

- KNN – 72%
- Random Forest – 77%
- Logistic Regression – 87%

The above resultant score is coming from columns and variables from the datasets.

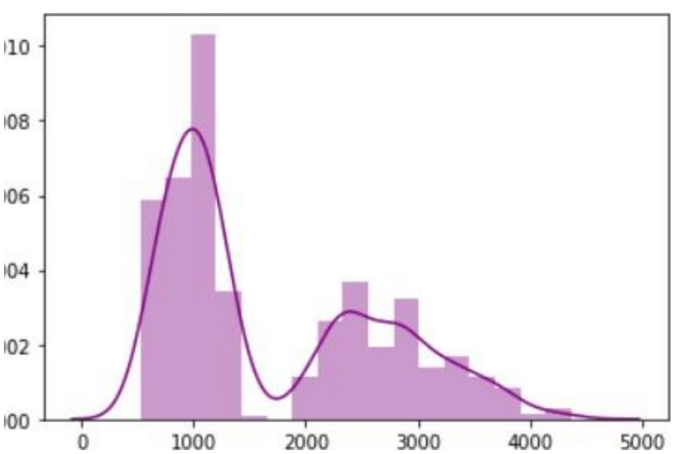


Fig 8.1 Variables

Fig 8.2 Independent Variable Dependency on Outcome

ENSEMBLE THE MODEL

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on the highest probability of chosen class as the output. It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on most of the voting. The idea is instead of creating separate dedicated models and finding the accuracy for each of them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

Voting Classifier supports two types of voting.

- **Hard Voting:** In hard voting, the predicted output class is the class with the most votes.
- **Soft Voting:** In soft voting, the output class is the prediction based on the average probability given to that class.

In the voting classifier the three algorithms KNN, Decision Tree, and Random Forest are passed and upon passing the data to the ensemble model the accuracy of the model is 81%

Fig 8.3 Ensemble Score

REFERENCES

[1] Wider pedestrian 2019. https://competitions. codalab.org/competitions/20132. 2, 3, 6, 7, 8

[2] Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Ab- Abhijit Ogale, and Dave Ferguson. Real-time pedestrian detection with deep network cascades. 2015. 3

[3] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In CVPR 2011, pages 3457–3464. IEEE, 2011. 3

[4] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu M Gavrila. Eurocity persons: A novel benchmark for person de- detection in traffic scenes. IEEE transactions on pattern analysis and machine intelligence, 41(8):1844–1861,2019. 3, 4, 5, 6, 7, 8

[5] Garrick Brazil, Xi Yin, and Xiaoming Liu. Illuminating pedestrians via simultaneous detection & segmentation. In Proceedings of the IEEE International Conference on Computer Vision, pages 4950–4959, 2017. 3

[6] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vas- concelos. A unified multi-scale deep convolutional neural network for fast object detection. In European conference on computer vision, pages 354–370. Springer, 2016. 3

[7] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos. Learning complexity-aware cascades for deep pedestrian de-detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 3361–3369, 2015. 3

[8] Zhaowei Cai and Nuno Vasconcelos. Cascade r-CNN: High-quality object detection and instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. 2, 4, 5, 6, 8

[9] Victor Campmany, Sergio Silva, Antonio Espinosa, Juan Carlos Moure, David Va´zquez, and Antonio M Lo´pez. Gpu-based pedestrian detection for autonomous driving.
Procedia Computer Science, 80:2377–2381, 2016. 1

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In international Conference on computer vision & Pattern Recognition (CVPR'05), volume 1, pages 886–893. IEEE Computer Society, 2005. 2, 3

[11] Piotr Dolla´r, Ron Appel, Serge Belongie, and Pietro Per- ona. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(8):1532–1545, 2014. 2

[12] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Per- ona. Pedestrian detection: An evaluation of the state of the art. IEEE transactions on pattern analysis and machine in- intelligence, 34(4):743–761, 2012. 2, 3, 4, 5, 6, 7, 8

[13] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In 2007 IEEE 11th in-

International conference on computer vision pages 1–8. IEEE, 2007. 3

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 3354–3361. IEEE, 2012. 3

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014. 3

[16] Hironori Hattori, Vishnu Naresh Boddeti, Kris M Kitani, and Takeo Kanade. Learning scene-specific pedestrian detectors without real data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3819– 3827, 2015. 1

[17] Amal Hbaieb, Jihene Rezgui, and Lamia Chaari. Pedes- train detection for autonomous driving within the cooperative communication system. In 2019 IEEE Wireless Communications and Networking Conference (WCNC), pages 1–6.
IEEE, 2019. 1

[18] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Gir- shick. Mask r-CNN. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017. 2, 4

[19] Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele. Taking a deeper look at pedestrians. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4073–4082, 2015. 3

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 2, 8

[21] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Bridging the gap between detection and tracking: A unified approach. In Proceedings of the IEEE International Conference on Computer Vision, pages 3999–4009, 2019. 1

[22] Jinpeng Li, Shengcai Liao, Hangzhi Jiang, and Ling Shao. Box-guided convolution for pedestrian detection. In

Proceedings of the 28th ACM International Conference on Multimedia, pages 1615–1624, 2020. 5, 6

[23] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017. 5
[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016. 2, 3
[25] Wei Liu, Irtiza Hasan, and Shengcai Liao. Center and scale prediction: A box-free approach for pedestrian and face de-detection. arXiv preprint arXiv:1904.02948, 2019. 5
[26] Wei Liu, Shengcai Liao, Weidong Hu, Xuezhi Liang, and Xiao Chen. Learning efficient single-stage pedestrian detect- tors by asymptotic localization fitting. In Proceedings of the
European Conference on Computer Vision (ECCV), pages 618–634, 2018. 3, 5, 6
[27] Wei Liu, Shengcai Liao, Weiqiang Ren, Weidong Hu, and Yinan Yu. High-level semantic feature detection: A new perspective for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 3, 5, 6, 7, 8
[28] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV), pages 181–196, 2018. 1
[29] Jiayuan Mao, Tete Xiao, Yuning Jiang, and Zhimin Cao. What can help pedestrian detection? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3127–3136, 2017. 3
[30] Stefan Munder and Dariu M Gavrila. An experimental study on pedestrian classification. IEEE transactions on pattern analysis and machine intelligence, 28(11):1863–1868, 2006. 3
[31] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton Van Den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In European Conference on Computer Vision, pages 546–561. Springer, 2014. 2
[32] Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Mask-guided attention network for occluded pedestrian detection, 2019. 3, 5
[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information pro- cessing systems, pages 91–99, 2015. 2, 3, 4, 5, 6
[34] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting humans in a crowd. arXiv preprint arXiv:1805.00123, 2018. 1, 3, 6, 7, 8
[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 5
[36] Xiaolin Song, Kaili Zhao, Wen-Sheng Chu Honggang Zhang, and Jun Guo. Progressive refinement network for oc-included pedestrian detection. In Proc. European Conference on Computer Vision, volume 7, page 9, 2020. 5, 6
[37] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. Fishnet: A versatile backbone for image, region, and pixel-level prediction. In Advances in Neural Information Processing Systems, pages 760–770, 2018. 2
[38] Paul Viola and Michael J Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004. 2
[39] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. arXiv preprint arXiv:1908.07919, 2019. 4, 5, 6, 8

[40] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: detecting pedestrians in a crowd. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7774–7783, 2018. 5
[41] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 794–801.
IEEE, 2009. 3
[42] Bo Wu and Ram Nevatia. Cluster-boosted tree classifier for multi-view, multi-pose object detection. In 2007 IEEE 11th International Conference on Computer Vision pages 1–8.
IEEE, 2007. 3
[43] Staining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500, 2017. 4, 5
[44] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In European conference on computer vision, pages 443–457.
Springer, 2016. 3

[45] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1112–1121, 2020. 1

[46] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. How far are we from solving pedestrian detection? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1259–1267, 2016. 3

[47] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3213–3221, 2017. 2, 3, 4, 5, 6, 7, 8

[48] Shifeng Zhang, Yiliang Xie, Jun Wan, Hansheng Xia, Stan Z Li, and Guodong Guo. Widerperson: A diverse dataset for dense pedestrian detection in the wild. IEEE Transactions on Multimedia, 2019. 1, 3, 6

[49] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 135–151, 2018. 3, 5

[50] ČSN 73 6110, "Designing urban roads," Prague: Czech normalization institute", 2006. [2] NZTA, 2007, "Pedestrian planning and design guide," "NZ Transport Agency", [Online] 2007 Available from: https://www.walk21.com/

[51] TP 189, "Determination of traffic intensity on roads (3rd edition)," 2018. [4] K. K. Liepmann, "The Journey to work," Kegan, Trench, Trubner, 1945. [5] T. Hägerstraand, "What about people in regional science?" Papers in regional science, vol. 24(1), pp. 7–24, 1970.

[52] M. E. Ben-Akiva, and S. R. Lerman, "Discrete choice analysis: theory and application to travel demand," MIT press, vol. 9, 1985.

[53] K. Kockelman, "Travel behavior as a function of accessibility, land use mixing, and land use balance: evidence from San Francisco Bay Area," Transportation Research Record: Journal of the Transportation Research Board, vol. 1607, pp. 116– 125, 1997.

[54]J. Wolf, "Using GPS Data Loggers To Replace Travel Diaries In the Collection of Travel Data," Dissertation, Georgia Institute of Technology, School of Civil and Environmental Engineering, Atlanta, GA, 2000.

[55] A. Banerjee, A. K. Maurya, and G. Lämmel, "A Review of Pedestrian Flow Characteristics and Level of Service over Different Pedestrian Facilities," Collective Dynamics, [S.l.], vol. 3, pp. 1-52, ISSN 2366-8539 [Online] 2018 Available from: https://collectivedynamics.eu/index.php/cod/article/view/A17, DOI: http://dx.doi.org/10.17815/CD.2018.17.

[56] Y. Oyama, and E. Hato. "Link-based measurement model to estimate route choice parameters in urban pedestrian networks." 2018.

[57] W. Zhu, H. J. P. Timmermans, "Modeling and simulating pedestrian shopping behavior based on principles of bounded rationality," In Timmermans HJP, editor, Pedestrian Behavior Models, Data Collection and Applications. Bingley: Emerald Group Publishing Ltd., pp. 137-156, 2009.

GitHub Link: Go for it