VIDEO AS A SENSOR: A CASE STUDY FOR BICYCLE RISKS

Ramakrishnan Narayanan
Department of Industrial and Enterprise Engineering
University of Illinois
104 S Mathews Ave, Urbana, IL 61801
Email: rnarynn3@illinois.edu

Kyle Begovich
Department of Mathematics
Department of Computer Science
University of Illinois
1409 W Green St, Urbana, IL 61801
Email: begovic2@illinois.edu

Fangyu Wu
Department of Civil and Environmental Engineering
Coordinated Sciences Laboratory
University of Illinois
205 N Mathews Ave, Urbana, IL 61801
Email: fwu10@illinois.edu

Richard Sowers, Corresponding Author
Department of Industrial and Enterprise Engineering
Department of Mathematics
University of Illinois
104 S Mathews Ave, Urbana, IL 61801
Email: r-sowers@illinois.edu

Daniel Work
Department of Civil and Environmental Engineering
Coordinated Sciences Laboratory
University of Illinois
205 N Mathews Ave, Urbana, IL 61801
Email: dbwork@illinois.edu

Word count (abstract): 147 words
Word count (text): 3882 words text + 10 tables/figures x 250 words (each) = 6382 words
Word count (references): 911 words

Submission Date

1st August 2017

ABSTRACT

In this article we consider the problem of assessing various road-related risks via sensor-enabled vehicles. With recent and substantial improvements in image processing software, it is now possible to consider large scale data analysis to aid in risk reduction strategies on roadways. To this end, we apply recent open-source neural-network image-processing software to investigate real-time and in situ classification of potential risk of bicycle injury. Namely, we identify bicyclists and identify whether they are wearing helmets or not. We present a proof-of-concept study and discuss the potential for widespread deployment of this and similar risk-quantification tools on sensor-enabled cars. Our study is based on a training set of about 500 images and leads to accuracy in excess of 60% on unseen images and confidence in excess of 80% on training set images.

Keywords: Video as a sensor, image classification, image segmentation, image processing, risk, bicycle safety

INTRODUCTION

Urban environments are undergoing a transitional change. Sensing, in part driven by the wide penetration of smartphones, is enabling a much better understanding of what is going on in cities. Mobile applications such as "WreckWatch" (1), "Realtime iOS Object Detection" (2), and pothole detection (3) are examples. Furthermore, the Controller Area Network (CAN) bus standard and the On-Board Diagnostics-II (OBD-II) system mean that vehicles themselves today have more sensors and more available data. Wide ranging, in-depth information is available, and many technologies already exist to take advantage of this for more than just vehicle diagnostics. This leads to vehicles, as a sensing platform, being able to analyze many aspects of an urban environment such as road or weather conditions (4). As this data is combined, it can lead to conjoint improvements and improved efficiencies in the decisions made by city inhabitants, city planners, and motorists. Video as a sensor is another major extension upon those ideas and will provide an even deeper understanding of urban environments and roadways.

Video is a particularly appropriate sensor for measuring and interpreting urban infrastructure. Driven by its presence in nearly all commercial cell phones, video capture has become incredibly inexpensive and is thus becoming ubiquitous. Video cameras are likely to become even more widespread as part of the array of sensors included on vehicles to enable autonomy. Vehicles can themselves become part of the suite of sensors in a city (4), (5). Already, many traffic lights are equipped with cameras. An even more dynamic and widespread coverage of the city is likely to arise as more and more cars are sensor-equipped. It is hoped that better decisions and traffic controls are likely to result (6).

An obvious use of commonplace video cameras is safety. Video surveillance is common for security reasons. Traffic light cameras help enforce observance of traffic signals. For the former, shoplifters and trespassers can be identified in hindsight. The latter is a bit more advanced; cars can be tracked as they cross an intersection and identified, using image recognition and license-plate identification software.

The focus of this effort is in understanding a bit more about the roadway and its environment. In particular, can one *quantify helmeted vs. unhelmeted bicyclists*? This is a case study intended to motivate more thinking about how vehicles-as-sensors can be used to understand risk. Unhelmeted cyclists are much more at peril in an accident (7). Studies have shown that both in sedans and SUVs, the impact speed of the bicyclists' head with the windscreen or hood, in case of a collision can be as high as 1.43 times the velocity of the vehicle (8), so the ability to identify and count the number of unhelmeted and helmeted cyclists could facilitate more precise risk assessments.

Problem Statement and Related Work

This initial bicyclist counting effort, carried out as a proof of concept, is inspired by a larger need to understand urban risk assessment using sensing via video. Vision Zero (9) is an international effort aimed at eliminating traffic-related casualties. In traffic accidents, pedestrians and cyclists are at most peril (10), so a valuable contribution would be to reliably count pedestrians and cyclists in various road-related environments. Pedestrian counting software is already well-developed (11) (12); our focus here is bicyclists. Cyclists with helmets are 85-90% less likely to receive a head injury from an accident than a cyclist without a helmet and 88% fewer accidents would result in a brain injury (13) (14) (15) (16). It would thus be of interest to separately count unhelmeted and helmeted cyclists at different times and in different regions of the city. This information may be useful in better quantifying this type of risk, and also may help safety initiatives to better target their efforts. This data might also, of course, help understand some meaningful policy implications

of risk programs (17) (in (18), a case is made that stricter helmet laws may, in fact, do more harm than good by reducing cycling than by reducing injuries).

Traditionally, camera-based object detection techniques rely on hand-crafted feature-based methods, such as Histograms of Oriented Gradients (HOG), coupled with Support Vector Machines (SVM's) to perform detection (19) (20). Recently, however, these techniques have taken a backseat to algorithms built on convolutional neural networks (21) (22). While it generally takes a significant amount of time and computational resources to train these neural networks, inference can be carried out much faster (23) (24). The versatility of currently available algorithms, coupled with better computational resources and hardware, is starting to them a viable solution for a number of real-time applications.

We here develop a bicyclist classifier using a recently-developed convolutional neural network based deep learning algorithm---"You Only Look Once" (YOLO) (25). This algorithm, as its name suggests, requires only one pass through an image, allowing it to be extremely fast. Counting helmeted vs unhelmeted cyclists provides a socially meaningful case study for exploring YOLO and its capabilities. Several technical aspects of this challenge are appealing:

- Identifying helmeted vs unhelmeted cyclists entails functionality which is more complex than identifying bicyclists alone. Moreover, the relevant feature of a helmet involves a small part of an image of a bicyclist. How well does YOLO do in understanding such refinements? While this task is in some ways more complex than simply identifying cyclists, it should, on the other hand, be a bit simpler to train YOLO since there are only two types of objects (helmet or not) of interest.
- How efficient and fast does YOLO work in this situation? One of YOLO's strengths is its speed. How well does that claim hold up in our test problem? One of the recurring problems in thinking about sensor-enabled vehicles is the speed with which one can process the large amounts of resulting data. Since we are only processing for one specific reason. what insight can be gained from this case study?
- What are the computational requirements to process the videos? Again, because of the vast amounts of data which video feeds can produce, sensor-enabled vehicles are likely to rely on both edge and cloud computing; can one locally process video feeds and send only the results to the cloud?

We hope that the specific application of this contribution will inform a larger discussion about using video as a sensor in more ways.

Outline and Contributions

This paper is organized as follows. The next section is dedicated to a summary of YOLO. We then summarize the training process and subsequently the training data. We follow this by a study of the performance, finding accuracy, even in this preliminary work, in excess of 60% on unseen images and confidence in excess of 80% on training set images. We end with a section on future work.

THE NEURAL NET

The edition of YOLO used for this project, YOLO version 2, is built from the ground-up with the "Darknet" architecture and is based on an interative analysis of very small subsets of the original image. It uses two types of operations; convolutional filters (which are matrix operations, and are also referred to as neurons or kernels) and max-pooling (i.e., finding a local maximum of a grid of

data), which acts as a nonlinear data reduction step. The model, Darknet-19 is made of 24 layers in total, of which 19 are convolutional and 5 are max pooling, and training consists of identifying the 'best' weights for the convolution steps; see Figure 1.

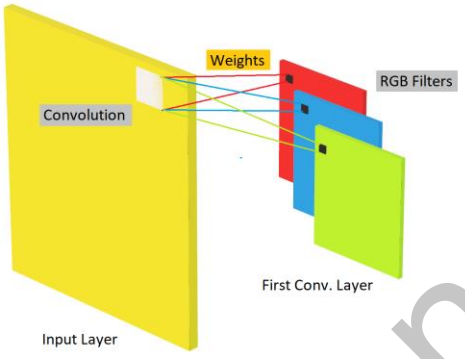

FIGURE 1 Filters and Convolutions

Yolo

Image processing algorithms designed to work with objects are built to perform two kinds of tasks: object recognition and object detection.

- Object Detection: The algorithm is designed to detect a specific type of object. Typically, object detectors have to look for the reference object at all positions and scales. If the required object is present in the image, the output is a location or bounding box around the required object (giving size and location of the object).
- Object Recognition: The input contains one of several objects at a known position and size, and the algorithm is built to decide which one is present.

Several previous image processing algorithms work as a hybrid between the two tasks. Object identification, where location, scale, and class are all sought, can be carried out by repurposing an object recognition algorithm, which can detect what class the object is but not if there is an object at that location. The recognition algorithm is then run over the image at multiple locations and scales, similar to what an object detection algorithm would do, in multiple iterations until the different object classes are listed, in combination with their locations and sizes. This method, being iterative, is slow. Redmon and Farhadi introduced YOLO in 2016 in (25), which is a convolutional neural network (CNN) implementation of an image processing algorithm built on the "Darknet" framework (26). YOLO is ideally suited for this project due to its capability to quickly and accurately process multiple images, as would be required for an image processing system on an autonomous vehicle.

FIGURE 2 The S × S grid division of an input image

YOLO is based on an algorithm that uses deep learning to predict object location and class. The image is first divided into an S × S grid to make regressors and classifiers, an example is shown in Figure 2 where S=12. When an input image is first processed by YOLO, it is divided into an S × S grid of cells (or sub-images) and a set of r multi-dimensional "regressors"

and C classifiers for each cell. A regressor is defined as a proposed bounding box. The number r of regressors can be tuned to optimize for the training and detection process over a range of possible bounding boxes (see (25) for a more detailed description). Each regressor gives five values (x,y,b,h,p) of interest: the x and y coordinates of the center of the bounding box relative to the center of the current cell, the breadth b and height h of the bounding box, and a confidence value $p \in [0, 1]$. The grid cell output also contains a classifier which predicts a conditional probability for each of C classes. The final output vector hence has a total number of dimensions equal to the number of all outputs from all cells, T:

$$T = (5 \times r + C) \times S \times S$$

Each cell is thus treated as a standalone system within the larger image. Each cell will generate r bounding boxes (with four coordinates and a confidence value). The heart of YOLO is the 24-layer neural network (19 convolutional layers and 5 max-pool layers) which give these vectors (x,y,b,h,p) for each regressor for each cell. The weights for the convolutional features between each layer is set during the training process.

Network structure

YOLO is designed to automatically work on a range of different images (25). YOLO downsamples each image into a 416 × 416 image, in order to have a 13 × 13 grid, where each grid cell is 32 pixels on a side. In Redmon's standard version, the object recognition model weights are pre-trained on the ImageNet dataset (27). The coordinates and classes used in object detection are predicted through "anchor box" priors (anchor boxes being randomly proposed bounding boxes) which efficiently suggest proposed bounding boxes. YOLO version 2 does not use a fully connected detection layer as in the case with most CNN's. It uses the anchor boxes to modify its predictions, as they are randomized prior predictions of bounding box outputs, obtained from k−means clustering on the training set. The priors were taken to be those that generated the best Intersection Over Union (IOU) scores, for the distance metric (25):

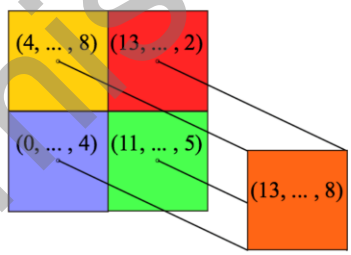

FIGURE 3 Maxpooling reduction step

$$d(box, centroid) = 1 - IOU(box, centroid)$$

Initially based on the GoogleNet (28) architecture, the edition of YOLO used for this project was motivated by the work on Network In Network (29). In (25), Redmon proposed the structure of 19 convolutional layers and 5 max-pool layers which is used here. Figure 3 shows an example of a max-pooling step. The number of filters is doubled after each max-pooling step to preserve the dimensionality of the required output.

Prediction Probability

In order to make predictions on the bounding box locations and the class of the object, YOLO uses simple probability. Each of the r regressors predicts one bounding box with a confidence value, which must, to make correct detections, be close to or equal to the value that is obtained through training, this value is hence the IOU of the ground truth and the proposed box. This confidence is a non-zero value if any object is predicted and zero otherwise. The confidence value is predicted as p:

$$p = Pr(object) \times IOU.$$

Here, Pr(object) is the probability that there is any object in that box. Further, the classifier within each cell of the grid also predicts the probability of the object being of a particular class, which is a conditional probability, assuming there are n possible classes: $\Pr(\text{class}_i|\text{object})\ \forall\ i \in (1, 2, \ldots, n)$. Then, using Bayes' theorem the expression can be rewritten in terms of Pr(object) as:

$$\Pr(class\ i\ |object) \times \Pr(object) \times IOU = \Pr(class\ i) \times IOU$$

for all $i \in \{1, 2, \ldots, n\}$. This overall confidence score thus incorporates class probabilities giving a value indicating how well the predicted box fits each of those n classes. Figure 4 below shows all of the "predicted boxes" that the regressors produce over each cell of the grid from the example image. After the all the predictions have been made, only those predictions that are above a certain threshold are output as true predictions. This is usually those predictions that are required, and a well-trained model will always provide predictions with conditional confidence values over the threshold only for the required predictions as shown in Figure 5.

FIGURE 4 All regressors predict bounding boxes. The 12 × 12 grid thus produces 144 × r bounding boxes in the grid.

SETUP

Training

Like other CNN based algorithms, YOLO works with small convolutions over the pixels in the input image. Filtering can be represented by a scalar-valued linear transformation and can be tuned to a specific feature. The filter is sequentially moved across the entire image. In image recognition tasks, the first convolution step usually has one filter for each of the RGB colors intensities. The weights in the filters are obtained through minimizing the error between the predicted value and the ground truth (the literature of neural networks provides a number of efficient ways to propagate error-minimization algorithms through multiple layers). This is an example of "supervised learning" where a predictor is developed after the algorithm goes through multiple images of being told where to find the object it is looking for (and in the process learning its features). This also gives us the ability to add any additional classifiers to YOLO. As mentioned, the training data was selected to represent a fairly broad collection of possible backgrounds, angles, lighting conditions, sizes, etc., and had to be closely annotated with the object locations, i.e., the ground truths. The parameters are adjusted to enable the algorithm to detect on unseen test images. Theoretically, for a set of weights to complete represent a "surface" that describes all features for a particular class (person, bicycle etc.), it would have to be trained on an infinite amount of data. Realistically, we sample a fraction of that data (which is our dataset) and try to extract relevant patterns; a richer training set should lead to better and more stable weights. YOLO is trained on a standard stochastic gradient descent (SGD) optimization, implying that as the weights are changed iteratively, there is a slow movement towards the optimum value (i.e., the minimum value of the loss function that is the change between the predicted and ground truth), with direction

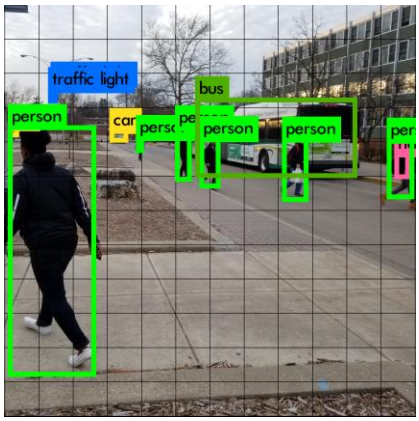

FIGURE 5 Bounding boxes that are retained after dropping those that have a confidence value lower than the threshold are the required boxes.

governed by the negative gradient of the cost function. Data augmentation techniques used in the training process replicate possible conditions when encountering new, unseen images. Random cropping of images during training gives the model the ability to work well in identifying objects even when they are partially covered by other obstacles or when there are multiple objects in planes that are parallel to each other. This is showcased in Figure 6(a). Further, augmentation techniques such as hue shifts, saturation and exposure shifts give it the ability to predict correctly in varying light conditions and backgrounds such as varying seasons, as well, as shown in Figure 6(b).

Data

YOLO's standard recognition set, as it currently stands, is already extremely comprehensive for transportation applications including, but not limited to: autonomous vehicles, traffic cameras, red light and toll cameras, and dash cams. Some of the included classifiers that have already been trained on include, but is not limited to, Person, Car, Truck, Bus, Train, Stop Sign, Stoplight, Motorbike, Bicycle. The pre-trained classifiers were developed using the ImageNet, Microsoft Common Objects in Context (COCO) (30) and PASCAL Visual Object Classes (VOC) (31) data sets.

TABLE 1 Video Properties

Property	Value
Width	1920
Height	1080
Codec	H.264(HP)
Framerate(frames per second)	30
Bitrate(kbps)	30584
Audio Codec	MPEG-4.AAC

To train on the additional class of helmeted cyclists, Neutral Cycle of Champaign of Champaign, IL supplied us with a series of videos a recent bicycle event. These videos were recorded on a GoPro Hero4 action camera. Most participants were helmeted. The video, in total, accounts for about 38000 frames. There were 30 frames per second, so much of the data was redundant. To give ample variation for the algorithm to train on, 548 images were manually selected; they were chosen to give a good mix of backgrounds and surroundings. Ten percent (10%) of these images were used as a validation set. The details of the videos are in Table 1.

FIGURE 6a Detections in bright light

FIGURE 6b: Detections with objects in varying planes

Performance

The training script took 24 hours to complete 10^4 iterations of stochastic gradient descent to find the weights for the regressors. The weights file gave usable results with considerably accurate predictions which predicted with confidence in excess of 60% on unseen images and confidence in excess of 80% on training set images. A brief tabulation of detection parameters is in Table 2.

TABLE 2 Detection Ratio and Times

Image	Objects Present	Objects Detected	Max Confidence	Detection Time (s)
1	3	2	0.68	10.4123
2	2	2	0.58	10.6919
3	1	1	0.75	10.4188
4	2	2	0.80	10.5459
5	3	2	0.79	10.4868
6	4	2	0.57	10.4377
7	2	1	0.75	10.4273

The training process was carried out on two fairly powerful NVidia M40 GPUs, but we believe better results can be achieved. The training, if carried out beyond the 24-hour run time would certainly give better results. More iterations and fine-tuning of the learning parameters could improve the detection speed and accuracy. During the training process, the IOU and Object Confidence values started by greatly differing between each other, to coming within 0.0133 of each other (where the values are in [0, 1]) with average recall rates going as high as 84% after 7500 iterations. It must also be taken into consideration that though the dataset was carefully chosen to reflect different conditions, colors and angles of the actual bicycle helmet, being an object with varying shape and color makes it difficult to properly train a classifier for, using 548 images. In comparison, the COCO dataset that trains for the "person" class has about 49000 images. The algorithm still performed exceptionally well on the validation set.

Our training effort in this case study focussed on identifying helmeted people. Unhelmeted bicyclists could thus be identified by detecting bicyclists whose bounding box did not overlap with that of a helmeted person. See Figure 7.

CONCLUSIONS AND FUTURE WORK

Our efforts indicate that video can reliably be used as a sensor using today's modern algorithms, which have excellent recognition capability. Even though we carried out training on a relatively small dataset, the results could be used to improve existing quantitative assessments of road hazard risks. Several related efforts are possible. Firstly, some more context might be available. If the bicyclist was really on the street, how close were they to the vehicle? Was there a bicycle lane, and if so, was the bicycle in the lane? More generally, are there other risks which could similarly be quantified; can we identify elderly people and children at intersections? As an even more refined question, can one identify unhelmeted children on bicycles (children bicyclists suffer the majority of serious head injuries in bicycling accidents (14)? Can we identify loss of driver attention (32)?

FIGURE 7 Box-location based context corroboration

We also note that once the weights of the neural network are computed, YOLO can be

implemented on an 'edge' computer, and the results can be uploaded to the cloud. This might address some of the privacy issues surrounding uploading full image streams to the cloud. There are also interesting vehicle-to-infrastructure or vehicle-to-vehicle questions. Can one vehicle signal to another 'watch out for the group of cyclists ahead' (33)?

We might also think about other classifications. The nature of our roadways is likely to change with more autonomous vehicles. There are in particular interesting questions about the behavior of traffic as autonomous vehicles and human-driven vehicles mix. Can we identify the proportion of autonomous vehicles on the road via video-as-a-sensor?

Automobile as a sensing platform.

Since sensing is a key component of autonomy, more and more cars in the future are likely to be equipped with video capabilities. In fact, the primary sensing platform of the future may be cars and trucks themselves. Embedding sensors in cars is more likely to lead to a significant portion of road vehicles having some of the most recent technologies. It may be more cost effective to rely on data from sensor-enabled vehicles than from infrastructure-dependent sensing (4).

Detection through visual obstructions.

Video capture gives a comprehensive view of the world in front of a driver, but where frame-by-frame detection is lacking is object permanence. While humans learn this skill in their early years, neural networks are generally not trained to detect in such a manner. The applications of this software, however, cannot have this shortcoming. Additions have been made within YOLO to track a detection, predict movement, and keep an object in scope even when a detection is not provided. Preparation to build upon this requires more research, including answers to questions including: what should the margin of error be for time without re-discovering an object, what objects are worthy of tracking, and how are driving decisions influenced by trackable objects versus non-trackable objects?

Acknowledgments: This material is based upon work supported by the Illinois Geometry Laboratory (IGL) and a seed grant from the Siebel Energy Institute. The authors would also like to acknowledge the invaluable assistance of Neutral Cycle of Champaign. We would like to thank Volodymyr Kindratenko of the National Center for Supercomputing Applications for time on his cluster. We would also like to thank the Idea Lab at the University of Illinois for their hospitality. Finally, this project would not have been possible without a large number of IGL scholars and graduate mentors. We would especially like to thank Eden Brewer and Dingyang Chen for their contributions to the paper. Other IGL students who have contributed to this effort are: Helen Babb, Yuanzhe Bian, Siqi Chen, Yijing Chen, Xinwei He, Sriram Krishnan, Yanning Li, Jingchi Liu, Yuhao Lu, Linzi Meng, Robert Monks, Sanskruti Bapurao More, Saumil Padhya, Yijia Qian, Bowen Song, Raphael Stern, Nilesh Thakkar, Mosaad Al Thokair, Yanbing Wang, Yizhu Wang, Wenting Xu, Yihan Zhang, Yihan Zhang, and Shuozhen Zhao.

REFERENCES

1. *Wreckwatch: Automatic traffic accident detection and notification with smartphones.* **White, Jules, et al.** 3, 2011, Mobile Networks and Applications, Vol. 16, p. 285.
2. **Young, Jay.** [Online] 2016. https://github.com/yjmade/ios_camera_object_detection.
3. *The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring.* **Balakrishnan, Jakob Eriksson and Lewis Girod and Bret Hull and Ryan Newton and Samuel Madden and Hari.** Breckenridge, U.S.A. : The Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008), June 2008.
4. *The Car as an Ambient Sensing Platform.* **Massaro, Emanuele, et al.** 1, 2017, Proceedings of the IEEE, Vol. 105, pp. 3--7.
5. *Urban Mobility in a Digital Age.* **Hand, Ashley.**
6. *Mapping the invisible: Street View cars add air pollution sensors.* **Google.**
7. *Bicycle helmets are highly effective at preventing head injury during head impact: Head-form accelerations and injury criteria for helmeted and unhelmeted impacts.* **Cripton, Peter A and Dressler, Daniel M and Stuart, Cameron A and Dennisona, Christopher R and Richards, Darrin.** Accident Analysis & Prevention, s.l. : Elsevier, 2014, Vols. 70 pp. 1-7.
8. *Pedestrian head impact dynamics: Comparison of dummy and PMHS in small sedan and large SUV impacts.* **Kerrigan, J. R and Arregui, C. and Crandall, J. R.** Stuttgart, Germany : 21st International Conference on the Enhanced Safety of Vehicles (ESV), 2009. 09-0127.
9. ***Vision Zero Network.***
10. ***National Highway Traffic Safety Administration.*** *Bicyclists and other cyclists: 2015 data. s.l. : National Center for Statistics and Analysis, Traffic Safety Facts, March 2017. DOT HS 812 382.*
11. *Pedestrian detection with deep convolutional neural network.* ***Chen, X., Wei, P., Ke, W., Ye, Q., and Jiao, J.*** *s.l. : In Asian Conference on Computer Vision, 2014, November. (pp. 354-365) Springer, Cham..*
12. *Monocular pedestrian detection: Survey and experiments.* ***Enzweiler, M. and Gavrila, D.M.*** *Transactions on pattern analysis and machine intelligence, s.l. : IEEE, 2009, Vols. 31(12), pp.2179-2195.*
13. *Effectiveness of bicycle safety helmets in preventing head injuries: a case-control study.* ***Thompson, Diane C, Rivara, Frederick P and Thompson, Robert S.*** *24, 1996, Jama, Vol. 276, pp. 1968-1973.*
14. *A case-control study of the effectiveness of bicycle safety helmets.* ***Thompson, Robert S, Rivara, Frederick P and Thompson, Diane C.*** *21, 1989, New England Journal of Medicine, Vol. 320, pp. 1361-136.*
15. *A case-control study of the effectiveness of bicycle safety helmets in preventing facial injury.* ***Thompson, Robert S, Thompson, Diane C and Rivara, Frederick P.*** *12, 1990, American Journal of Public Health, Vol. 80, pp. 1471-1474.*
16. *Do bicycle safety helmets reduce severity of head injury in real crashes?* ***Dorsch, et al.*** *3, 1987, Accident Analysis & Prevention, Vol. 19.*
17. *Can injury prevention efforts go too far?: Reflections on some possible implications of Vision Zero for road accident fatalities.* ***Rune, Elvik.*** *3, 1999, Accident Analysis & Prevention, Vol. 31, pp. 265--286.*
18. *Head injuries and bicycle helmet laws.* ***Robinson, Dorothy L.***
19. *Vision-based bicycle detection and tracking using a deformable part model and an EKF algorithm.* ***Cho, H., Rybski, P.E. and Zhang, W.*** *s.l. : IEEE, In 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), September 2010. pp. 1875-1880.*
20. *Vision-based bicyclist detection and tracking for intelligent vehicles.* ***Cho, H., Rybski, P.E. and Zhang, W.*** *s.l. : IEEE, In Intelligent Vehicles Symposium (IV), 2010., June 2010. pp. 454-461.*
21. *Rich feature hierarchies for accurate object detection and semantic segmentation.* ***Girshick, R., Donahue, J., Darrell, T. and Malik, J.*** *s.l. : IEEE, In Proceedings of the IEEE conference on computer vision and pattern recognition., 2014. pp. 580-587.*
22. *Imagenet classification with deep convolutional neural networks.* ***Krizhevsky, A., Sutskever, I. and Hinton.*** *s.l. : G.E., Advances in neural information processing systems, 2012. pp. 1097-1105.*
23. *Faster R-CNN: Towards real-time object detection with region proposal networks.* ***Ren, S., He, K., Girshick, R. and Sun, J.*** *s.l. : Advances in neural information processing systems., 2015. pp. 91-99.*

24. Ssd: Single shot multibox detector. ***Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C.*** *s.l. : Springer, Cham. European conference on computer vision., October 2016. pp. 21-37.*

25. You only look once: Unified, real-time object detection. ***Redmon, Joseph, et al.*** *2016, Vol. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788.*

26. ***Redmon, Joseph.*** *Darknet: Open Source Neural Networks in C. 2013-2016.*

27. ImageNet Large Scale Visual Recognition Challenge. ***Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg and Li Fei-Fei.*** *s.l. : Springer Netherlands, December 2015. International Journal of Computer Vision. Vol. 115, pp. 3 211-252. 0920-5691.*

28. ***Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.*** *Going Deeper with Convolutions. September 2014. arXiv:1409.4842v1 [cs.CV].*

29. ***Lin, M., Chen, Q., and Yan, S.*** *Network In Network. December 2013. arXiv:1312.4400v3 [cs.NE].*

30. Microsoft COCO: Common Objects in Context. ***Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L.*** *s.l. : CoRR, 2014, Vol. abs/1405.0312.*

31. ***Everingham, M., Van-Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A.*** *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. s.l. : http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.*

32. Driver Gaze Region Estimation without Use of Eye Movement. ***Lex, Fridman, et al.*** *3, 2016, IEEE Intelligent Systems, Vol. 31, pp. 49-56.*

33. Dedicated short-range communications (DSRC) standards in the United States. ***Kenney, John B.*** *7, 2011, Proceedings of the IEEE, Vol. 99, pp. 1162-1182.*