Project Title: Pedestrian detection at night with Deep Learning

Authors

Olivier Nicolini oliviern@kth.se

Simone Zamboni zamboni@kth.se

Abstract

Pedestrian detection at night is one of the key challenges for self-driving cars. To tackle this problem the NightOwls dataset, a new dataset containing images of pedestrians at night, has been released. We decided to explore this dataset by investigating the best ratio between pedestrian to background images to train a Fast R-CNN network and if there are image improvement techniques from computer vision that can be used to improve pedestrian detection. We found out the best ratio is a 50-50 split between pedestrian and background images and that adaptive gamma correction can improve the detection of pedestrians at night.

Introduction

There is an increasing interest in autonomous cars which rely on many sensors such as cameras, lidars and depth sensors in order to move in space. How well cars are able to exploit these sensors to see and move is not only important for moving from point A to B but also to detect if obstacles such as pedestrian are present. Precise and reliable pedestrian detection on the streets is one fundamental blocks for creating safe self-driving vehicles.

After the recent developments of Deep Learning (DL), automatic detection of objects in images has become a viable option, thanks to architectures like Fast R-CNN [8], Faster R-CNN [9] and YOLO [25].

These types of deep learning models are also used in cameras for self driving cars. At night, unfortunately, autonomous cars cannot rely solely on detection models with cameras images since detection with dark images has proven to be more difficult than with bright images (with Caltech [19] achieved 7.36% miss rate [11], with NightOwls 17.8% [18]). Therefore, one of the key challenges in the development of these vehicles is a reliable night vision algorithm to enable to see also in the dark in order to ensure safer streets for everyone.

To help the scientific community, we tried to understand which improvements can be made to ameliorate the training and the performance of deep learning models in order to give guidelines for future developments of detection models made for dark images. To do that, we decided to use Faster R-CNN [9], which is considered to be the state-of-the-art for detection models, and analyze the performance of this algorithm on the *NightOwls* dataset [18], a dataset specifically made for pedestrian detection, containing images of pedestrians taken at night from a car. The dataset contains 270k images of which only 32k contain at least one instance of pedestrians.

In this paper, we explore which is the best combination of images containing instances of pedestrian and of images without them (which we call *background images*) to be fed to the model during training, in the case of the NightOwls dataset. This could be useful for improving the training of object detection networks by balancing the training set and also for future research to create more balanced datasets for pedestrian detection. In the literature, we did not find any studies that address in detail the split between background and annotated images in object detection using neural networks, so this is our first contribution.

To further improve the performance of Faster R-CNN, we decided to use different types of preprocessing such as histogram equalization [1], adaptive gamma correction [6], multiscale retinex algorithm [2], and Ying's image contrast enhancement [3] to modify the image before entering the model and test if these modifications led to improvements in detection. This is our second contribution, proving that designing a preprocessing and that train a network on preprocessed images can be better than the typical end-to-end learning neural networks are capable of.

Theoretical framework/literature study

Computer Vision is one of the most active fields in ICT, and innovations in it are driven by both new methods and new datasets available.

One of the most important datasets in this field has been ImageNet [13], and since 2012 when AlexNet [12] achieved a lower error rate than any other previous attempt, deep neural networks are the main tool used in computer vision.
AlexNet was built to tackle the classification problem, but tasks in computer vision are multiple and often more challenging. One of the most important one is object detection, that became more accessible thanks to both neural networks developments and new datasets specifically conceived for detection like Pascal VOC [21].

One of the first deep architecture that tried to tackle the Pascal VOC dataset was R-CNN [20], which utilizes a region proposal algorithm to extract the interesting regions in the image, then a deep convolutional neural network (CNN) to extract the features from every proposed region and an support vector machine (SVM) to classify the features of the proposed region.

This first architecture reached new state-of-the-art results but it was inefficient and very slow (47 seconds for each test image), so this led to the development of Fast R-CNN [8]. This new architecture, instead of feeding the proposed regions to the CNN, feeds directly the input image to a convolutional network and then the region proposal algorithm extracts the interesting regions to be classified in a fully connected (or dense) neural network. This means that the image is passed only once in the convolutional neural network, and therefore the training and test time of this network is a great improvement over R-CNN.

In order to optimize even more the performance on the object detection task Faster-R-CNN [9] was developed. In this architecture, the image is fed as an input to a convolutional network which outputs a feature map. Instead of using a region proposal algorithm on the feature map, a separate network is used to predict the regions. These regions are then reshaped and used to classify the image class and to predict the offset values of the bounding box.
Since then only variations of Faster-R-CNN achieved state-of-the-art performances and now this architecture is the baseline for object detection, even if other architecture can achieve good results with less computation time (see YOLO [25]).

The improvements done were driven not only by better architectures but also thanks to new datasets, especially the COCO dataset [16], that consists (in its latest iteration) of more than 330K images (>200K labelled) and 1.5 million object instances with 80 object categories.

The COCO dataset contains images of different domains (persons, vehicles, animals…), while other datasets specialize in images in a more restricted domain. In pedestrian detection, the most used dataset are Caltech [19], KITTI [24], CityPersons [4] and KAIST [17]. This last one is the only one containing night images, because 5 out of 10 recordings had been done during the night, but all the recording happened in a single city. Among these datasets, the most commonly used one is Caltech, because it contains 250k images and 280k pedestrian boxes splitted roughly equally between training and test, even if the images are only of size 640x480.
Together with these new datasets also architectures designed specifically for pedestrian detection have been released, such as Checkerboards [10], Adapted Faster R-CNN [4], RPN+BF [14] and SDS-RCNN [11].

Our work is instead based on the NightOwls [18] dataset, which contains images of pedestrians at night with annotated bounding boxes. The dataset contains 279k fully annotated frames with 42k pedestrians, where 32k frames contain at least one annotated object and the remaining 247k are background images. The images were captured in 7 European cities around 3 countries (Germany, Netherlands and UK) using a camera with image resolution of 1024 x 640 at a frame rate of 15fps.

In total 40 individual recordings were captured. The annotations for the test set are not available since they have been used for a challenge [29].

In the past, the standard approach to deal with dark images was to use different types of preprocessing such as gamma correction [6], histogram equalization[1], changes in brightness and in the colour space of the image. Here we will analyze some of the most relevant for our purposes.

Gamma correction (GC) is used to correct the luminance of an image. A more advanced version of this function is brought by *adaptive gamma correction* (AGC) [6] that dynamically determines the right intensity transformation function based on the input image characteristics.

Histogram Equalization (HE) is a method that is used to calibrate the contrast of an image using its histogram. This is done by spreading the most frequent intensity values in an image allowing an overall better distribution of intensity in the image. A more advanced version of this technique is AHE (adaptive histogram equalization) [1] where multiple histograms corresponding to different parts of an image are calculated used to redistribute the lightness values of the image. This method usually also increases the noise of an image, therefore, to decrease this effect *Contrast Limited AHE* (CLAHE) [7], a variant of adaptive histogram equalization, has been developed. This method limits contrast amplification to prevent noise amplification.

Another approach has been brought by the use of *retinex algorithm* [2] to improve the quality of the image. The basic idea of the Retinex algorithm is based on Land’s theory of image perception [22] and works by separating illumination from the reflectance in a given image. This is done in 2 steps: first, the algorithm does an estimation and then a normalization of illumination. Among the different implementation of the retinex algorithm, one of the best is the *multiscale retinex with colour restoration* (MSRCR) [2] algorithm that combines colour constancy with local contrast enhancement in order to render images more similar to how it is believed humans perceive objects.

One of the latest developed algorithms is Ying’s *image contrast enhancement algorithm* [3] that uses exposure fusion on multi-exposure images that are synthesized from an input image.

A new approach to enhance dark image has been brought by deep learning models specifically made for brightening dark images as in Chen’s *Learn to See in the Dark* [15] model which is trained using images of the same scene in dark and bright light exposure. Unfortunately, this model only works on the raw images taken directly from the camera, that in our case are not available.

Research questions, hypotheses

Our hypothesis is that deep learning architectures can be used to analyze RGB cameras images and correctly detect pedestrians in dark environments and that preprocessing on these images can be used to further improve the performance of such models.

By using Faster R-CNN trained on the NightOwls dataset as a baseline model, we will find the optimal rapport of background-only images and images with pedestrian for training the network.

We will then proceed to assess if different types of image preprocessing are enough to improve the performance of current methods.

Method(s)

Various experiments were conducted to find out the answers to our research questions and have quantitative data collected to show the best approach in each of the settings.

In every experiment, an implementation of Faster R-CNN [9] was used with a ResNet101 [5] pre-trained on ImageNet [13] as the backbone and the rest of weights pre-trained on the COCO dataset [16]. The learning rate was set to 1e-3, with a batch size of two images and SDG as the optimizer. At test time only the predicted bounding boxes with confidence greater than 0.7 were considered as valid predictions. We decided to use Faster R-CNN because it has shown to perform accurately in various domains and it is well known in the community, so our results findings can be more easily applied.

To assess the performance of our model we used mAP and average miss rate, two popular metrics in object detection, the first used to assess the performances on the COCO dataset and the second one is used in the NightOwls dataset paper [18].
mAP, or *mean average precision*, is the mean of the average precision scores for each query and is formally defined as:

$$mAP = \frac{1}{N} \Sigma_{i=1}^{N} AP_i$$

where AP is the average precision for query i.

Average precision is the area under the precision-recall curve and is formally defined as:

$$AP = \int_0^1 p(r)dr$$

where p(r) is the precision as a function of recall r.

The miss rate (MR) can be defined as:

$$MR = \frac{FN}{FN + TP}$$

where FN is the number of false negatives and TP is the number of true positives.

Due to the computational resources available to us, we cannot compete on the miss rate achieved on the NightOwls dataset paper, but we can do some smaller-scale experiments and find out the most effective way to train a network on this dataset.

We found some incongruencies between the data reported on the paper and the data reported on the official website since on the official site of the dataset[32] it is written that the images were captured with a frame-rate of 5,33fps by taking every third frame with an original frame-rate of 16fps. In order to better utilize our resources and reduce the number of training images only one image out of two was taken into consideration when creating our training set, effectively reducing the frame rate to 2.665fps. Therefore, we considered only 65k images out of the original 130k. Thus, the images containing pedestrians we considered are 12k, instead of the original 24k. We believe that this modification does not influence the performance of the network because at 5.33fps an image is captured every 0.188s and at 2.655fps an image is captured every 0.376s, so still a relatively short time between one image and the other.

For the first round of experiments whose goal was to find the best pedestrian-background images, the training set used was the following:

- 12k images with pedestrians (the NightWols dataset training set has 128k images, but only 24k have at least one pedestrian in it and therefore our subset had only 12k images containing pedestrians), so 100% of the images were with pedestrians;
- 9k images with pedestrians and 3k background images. These images were chosen randomly in a way to have a split 75% pedestrians and 25% background
- 6k images with pedestrians and 6k background images. These images were chosen randomly in a way to have a split 50% pedestrians and 50% background

- 3k images with pedestrians and 9k background images. These images were chosen randomly in a way to have a split 25% pedestrians and 75% background

We trained the network for three epochs on each of these settings and then we evaluated it on the whole NigthOwls validation set. Results are reported in the next section. The metrics for the precision and recall are generated using the official Python COCO API[30] and the miss rate can be computed using the code on the online challenge for the NightOwls dataset[29].

The second round of experiments was aimed to find a pre-processing that improved the performance of the network. We choose the best split among the previous settings and we applied the preprocessing to the images using the CPU before the images were loaded in the GPU and fed into the network. This process was also done in the testing phase.

The image processing techniques used are:

- Histogram equalization(HE), using the implementation at [27]
- Automatic Gamma Correction(AGC), using the implementation at [26]
- CLAHE, using the implementation at [27]
- MSRCR, using the implementation at [33]
- Ying 2017, using the implementation at [31]

Examples of these techniques can be seen in Figure 1, Figure 2 and Figure 3.

All the experiments were carried out on Google Cloud Platform[28] using virtual machines with 4CPUs, 16GB of RAM and an Nvidia K80 as GPU. Experiments without preprocessing took around 36 hours to complete, of which 24 for training and 12 for testing.

Results and Analysis

The authors of the NightOwls dataset made available online [29] the predictions of the Faster R-CNN mentioned in the dataset paper. The network was trained for 100k iterations on the whole dataset (how many images in each iteration is unknown) and uses VGG16 [23] as the backbone.
We used these results to compare them to our results, obtained by training on 12k images for three epochs with a batch size of 2, so totalling 18k iterations.

Table 1 reports the results of the first round of experiments, aimed to find the best ratio between pedestrian images and background only images.

Pedestrian images in the training set	100%	75%	50%	25%	Faster R-CNN NightOwls
Average Precision (AP) with IoU=0.50:0.95 (official COCO benchmark)	**0.009**	**0.023**	**0.051**	**0.041**	**0.060**
Average Precision (AP) with IoU>0.50	0.021	0.054	0.105	0.090	0.134
Average Precision (AP) with IoU>0.75	0.005	0.016	0.040	0.028	0.038
Average Precision (AP) with IoU=0.50:0.95 and area = small	0.004	0.08	0.018	0.013	0.019
Average Precision (AP) with IoU=0.50:0.95 and area = medium	0.010	0.021	0.052	0.040	0.064
Average Precision (AP) with IoU=0.50:0.95 and area = large	0.009	0.053	0.095	0.082	0.089
Average Recall (AR) with IoU=0.50:0.95	0.016	0.042	0.075	0.058	0.084
Average Recall (AR) with IoU=0.50:0.95 and area = small	0.010	0.017	0.031	0.023	0.023
Average Recall (AR) with IoU=0.50:0.95 and area = medium	0.018	0.041	0.078	0.058	0.089
Average Recall (AR) with IoU=0.50:0.95 and area = big	0.011	0.081	0.115	0.107	0.128
Average Miss Rage (AMR) when bounding box height > 50px (official NightOwls benchmark)	**0.872**	**0.691**	**0.479**	**0.552**	**0.291**
Average Miss Rage (AMR) small (bounding box height between 50 and 75 px)	0.851	0.781	0.627	0.725	0.410
Average Miss Rage (AMR) with heavy occlusion when bounding box height > 50px	0.901	0.838	0.714	0.744	0.823
Average Miss Rage (AMR) on all bounding boxes	0.878	0.729	0.546	0.611	0.428

Table 1. Results of the first round of experiments.

Figure 1. Different preprocessing confronted on an image of the NightOwls dataset.

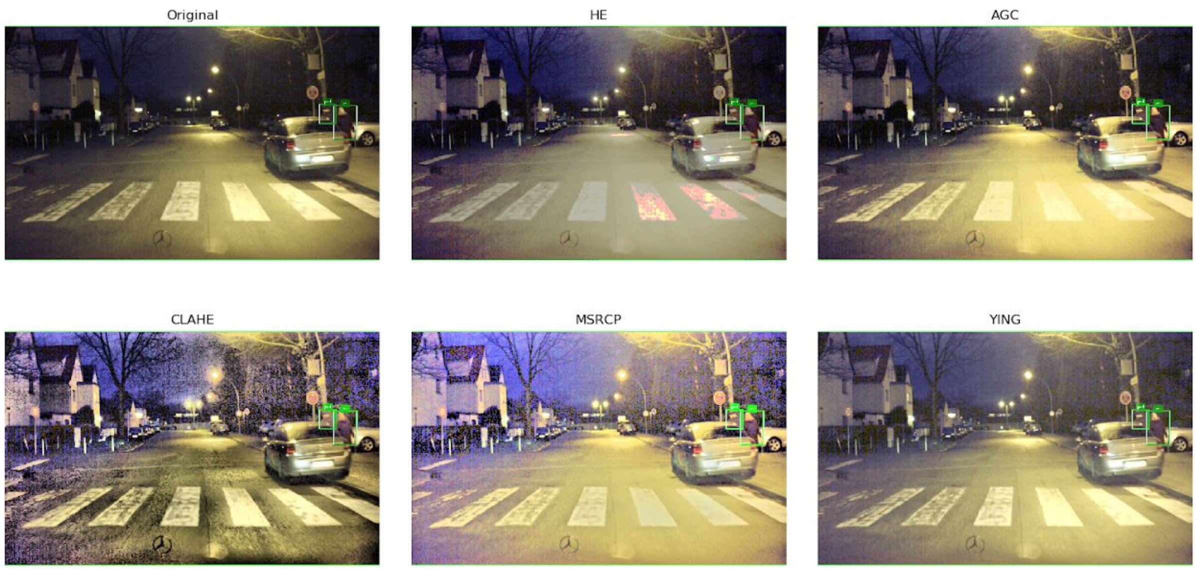

Figure 2. Different preprocessing confronted on an image of the NightOwls dataset.

Figure 3. Different preprocessing confronted on an image of the NightOwls dataset.

For the second round of experiments, we kept the 50-50 split in the training set since it was the one produced better results and we applied the various preprocessing to the images. Results can be found in Table 2.

Histogram equalization. Preprocessing the images with histogram equalization increased the computational time by a negligible time, in fact, we cannot see any difference in the training time. Unfortunately, this type of preprocessing distorted too much the images and therefore the network was not able to converge and to produce any result.

CLAHE. CLAHE increased the training time by around 20%, but as with the standard histogram equalization this technique distorted too much the images and therefore the network was not able to converge and to produce any result.

Adaptive Gamma Correction. This algorithm increased the training time by 10%, but the network was able to converge and produced better results than any of the previous experiments; results are in Table 2.

MSRCR. This type of preprocessing increased the training time by five times. This is may due to the fact that the CPUs used were not very powerful, but still, an increase of the training time by five times makes impractical the use of such method.

Ying 2017. This algorithm increased the training time by four times. This is may due to the fact that the CPUs used were not very powerful, but still, an increase of the training time by this much makes impractical the use of such method.

Preprocessing	Adaptive Gamma Correction	Fast-RCNN NightOwls
mean Average Precision (mAP) with IoU=0.50:0.95 (official COCO benchmark)	**0.058**	**0.060**
mean Average Precision (mAP) with IoU>0.50	0.121	0.134
mean Average Precision (mAP) with IoU>0.75	0.044	0.038
mean Average Precision (mAP) with IoU=0.50:0.95 and area = small	0.025	0.019
mean Average Precision (mAP) with IoU=0.50:0.95 and area = medium	0.059	0.064
mean Average Precision (mAP) with IoU=0.50:0.95 and area = large	0.090	0.089
Average Recall (AR) with IoU=0.50:0.95	0.101	0.084
Average Recall (AR) with IoU=0.50:0.95 and area = small	0.055	0.023
Average Recall (AR) with IoU=0.50:0.95 and area = medium	0.106	0.089
Average Recall (AR) with IoU=0.50:0.95 and area = big	0.132	0.128
Average Miss Rage (AMR) when bounding box height > 50px (official NightOwls benchmark)	**0.381**	**0.291**
Average Miss Rage (AMR) small (bounding box height between 50 and 75 px)	0.524	0.410
Average Miss Rage (AMR) with heavy occlusion when bounding box height > 50px	0.656	0.823
Average Miss Rage (AMR) on all bounding boxes	0.467	0.428

Table 2. Performance of the 50-50 split with added adaptive gamma correction against the baseline Fast RCNN of the NightOwls dataset authors.

Discussion

From the results of our experiments, it is clear that the best combination of training images was the 50-50 split between pedestrian images and background images. This is probably the case since with only two classes (pedestrian and background) the network needs also some background images to avoid proposing too many false positives during test. Interestingly, the 25-75 split performed better than the 75-25 split, even if it was trained with three times more pedestrians.

In the original training set, the NightOwls dataset contained 130k images, of which only 24k with pedestrians. This means that the ratio between background images over total images is around 0.82. Our research points out that this high percentage of background images might not useful.

Therefore we suggest a new practice to be used for object detection datasets with a lot of background images: to not simply train a network using all the available images but experiment with different ratios of images with annotations and background images. It may be the case that a dataset contains too much background images and therefore the network struggle to propose bounding boxes, or it could be the case that the dataset has too little background images and then when the network is tested it generates too many false positives. Therefore an optimal ratio for the detection task on each dataset could greatly improve the performance. It could be possible to obtain better performance with less data and training time, therefore we suggest the scientific community to also consider the ratio between annotated and background images when training neural networks for object detection.

Of course, the reported findings could also be useful when deciding how much and which type of data to gather for a new pedestrian dataset. By knowing what has been discussed here, the authors of the NightOwls dataset could have tried to gather images from more crowded places in order to have more images with pedestrians, instead of having 82% of the images without them. If an organization is planning to gather and annotate pedestrians images, this work could give insights on how and where to gather this data, and how the balance between background and annotated images should be.

Our work also demonstrates that if images are taken in difficult environments, they can be first improved and then fed to a neural network, instead of simply relying on the end-to-end learning of neural networks. This works also shows that this is not a trivial task, since aspects like computational time and distortion should also be taken into account. A viable preprocessing method should not increase the computational time too much such that it becomes more convenient to simply train more a network without modifying the original images. This is the reason why we believe computationally heavy methods like MSRCR or Ying are not practical. In the case of the NightOwls dataset adaptive gamma correction is an excellent trade-off between improvement and performances, since it only increased computation time of around 10% but it generated great improvements.

By training a Faster R-CNN for 18k iterations we achieved an mAP of 0.058 compared to the Faster R-CNN trained for 100k iterations by the authors of the NightOwls dataset with an mAP of 0.060. Therefore we can confidently assume that optimizing the ratio between background and annotated images and improving the images with algorithms such as adaptive gamma correction can greatly improve detection performances.

References

[1]O. Patel, Y. P. S. Maravi, and S. Sharma, “A Comparative Study of Histogram Equalization Based Image Enhancement Techniques for Brightness Preservation and Contrast Enhancement” SIPIJ, vol. 4, no. 5, pp. 11–25, Nov. 2013.

[2]D. J. Jobson, Z. Rahman, and G. A. Woodell, “A multiscale retinex for bridging the gap between color images and the human observation of scenes” IEEE Trans. on Image Process., vol. 6, no. 7, pp. 965–976, Jul. 1997.

[3]Z. Ying, G. Li, Y. Ren, R. Wang, and W. Wang, “A New Image Contrast Enhancement Algorithm Using Exposure Fusion Framework” in Computer Analysis of Images and Patterns, vol. 10425, M. Felsberg, A. Heyden, and N. Krüger, Eds. Cham: Springer International Publishing, 2017, pp. 36–46.

[4]S. Zhang, R. Benenson, and B. Schiele, “CityPersons: A Diverse Dataset for Pedestrian Detection” arXiv:1702.05693 [cs], Feb. 2017.

[5]K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition” arXiv:1512.03385 [cs], Dec. 2015.

[6]S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, “Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution” IEEE Trans. on Image Process., vol. 22, no. 3, pp. 1032–1041, Mar. 2013.

[7]P. D. Ferguson, T. Arslan, A. T. Erdogan, and A. Parmley, “Evaluation of contrast limited adaptive histogram equalization (CLAHE) enhancement on a FPGA” in 2008 IEEE International SOC Conference, Newport Beach, CA, USA, 2008, pp. 119–122.

[8]R. Girshick, “Fast R-CNN” arXiv:1504.08083 [cs], Sep. 2015.

[9]S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks” arXiv:1506.01497 [cs], Jan. 2016.

[10]S. Zhang, R. Benenson, and B. Schiele, “Filtered Channel Features for Pedestrian Detection” arXiv:1501.05759 [cs], Jan. 2015.

[11]G. Brazil, X. Yin, and X. Liu, “Illuminating Pedestrians via Simultaneous Detection & Segmentation” arXiv:1706.08564 [cs], Jun. 2017.

[12]A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks” Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2012.

[13]J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database” p. 8.

[14]L. Zhang, L. Lin, X. Liang, and K. He, "Is Faster R-CNN Doing Well for Pedestrian Detection?" arXiv:1607.07032 [cs], Jul. 2016.

[15]C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to See in the Dark" arXiv:1805.01934 [cs], May 2018.

[16]T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context" arXiv:1405.0312 [cs], Feb. 2015.

[17]S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline" in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1037–1045.

[18]L. Neumann et al., "NightOwls: A Pedestrians at Night Dataset" in Computer Vision – ACCV 2018, vol. 11361, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 691–705.

[19]P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art" IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, pp. 743–761, Apr. 2012.

[20]R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation" arXiv:1311.2524 [cs], Oct. 2014.

[21]M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge" Int J Comput Vis, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[22]E. H. Land, "The Retinex Theory of Color Vision" Sci Am, vol. 237, no. 6, pp. 108–128, Dec. 1977.

[23]K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition" arXiv:1409.1556 [cs], Apr. 2015.

[24]A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset" The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[25]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection" arXiv:1506.02640 [cs], May 2016.

[26] qyou, AGCWD, GitHub, https://github.com/qyou/AGCWD/blob/master/agcwd.py

[27] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) https://doi.org/10.7717/peerj.453

[28]Google Inc, Google Cloud, https://cloud.google.com/

[29] nightowls-competition@googlegroups.com, NightOwls Pedestrian Detection Challenge 2019, https://competitions.codalab.org/competitions/21231#participate

[30] COCO Consortium,COCO API, GitHub, https://github.com/cocodataset/cocoapi

[31] AndyGuang, Image-Contrast-Enhancement, GitHub, https://github.com/AndyHuang1995/Image-Contrast-Enhancement

[32] nightowls-competition@googlegroups.com, NightOwls dataset website, http://www.nightowls-dataset.org/nightowls-competition-2019/

[33] Zheng Liu, Retinex, GitHub, https://github.com/dongb5/Retinex/