
Self-supervised Sun Glare Detection CNN for Self-aware Autonomous Driving

Yiqiang Chen

RAMS Reliability Technology Lab
Huawei Technology Co., Ltd.
chenyiqiang@huawei.com

Feng Liu, Ke Pei

TTE-DE RAMS Lab
Huawei Technology Co., Ltd.
{feng.liu1, peike}@huawei.com

Abstract

For the fully autonomous driving systems that need to work under any circumstances, it is essential to detect if there is any degradation of the perception models and be aware of the robustness against the different weather conditions. Sun glare has always been an issue for manual driving and is becoming a real problem for autonomous driving as well. Ignoring and letting the algorithms work on glare-corrupted camera images can lead to fatal consequences. In order to achieve self-awareness, in this paper, we propose a sun glare detection approach and robustness benchmark to sun glare corruption based on glare rendering. In the benchmark, different levels of glare are introduced to assess the vulnerability of CNN detectors. With the help of self-supervised learning, our detection approach tackles the problem of glare data collection and annotation. Online glare synthesizing allows the CNN to take various and diverse training data, which makes the model robust and easy to generalize. We experimentally show that our method outperforms the state-of-the-art methods.



Figure 1: Sun glare can cause a safety hazard for both manual and autonomous driving. (a) Glare causes temporal blindness to the driver. (b) The glare obstructs the front vehicle. This leads to a partial loss of environment perception and consequently endangers the overall safety of the automated vehicle.

1 Introduction

Challenging weather conditions are the primary safety hazards for both manual and autonomous driving. The low-lying sun with overly intense light can gravely affect the driver’s visibility or the visual sensors of an automated vehicle, and it causes temporary blindness. This makes the events like car breaking, speed-limit change or pedestrian difficult to spot and further leads to perilous situations. Some studies show that the presence of sun glare is associated with traffic accidents. Redelmeier et al. [1] found that the chance of car crashes under bright sunlight is about 16% higher than under

normal weather. Moreover, Hagita et al. [2] showed that sun glare significantly increases traffic accidents in bicycles and pedestrians encountering turning vehicles.

When a camera’s view gets obstructed by sun glare, the perception performance of the autonomous driving system can be degraded and lead to subsequent failures. This exposes the vehicle’s intended functional and overall safety to great dangers and can even be life-threatening for passengers in the vehicle and other road users. Regarding this matter, it is crucial that autonomous systems perform self-monitoring and self-assessment to the perception to evaluate if the system is still capable of achieving its pre-defined function at an expected level.

Despite the importance of glare detection as an autonomous driving perception self-check function, there is a very limited number of previous works on this task with cameras in the state of the art. Most of the existing methods leverage the image properties, especially saturated brightness and extract affected regions via image processing heuristics like thresholding. However, these approaches easily produce false detections since they cannot distinguish sun glare from other bright regions on the image (ex. bright sky, white cloud). In consideration of the stability of the system, it is necessary to use a more robust glare detector as the monitoring function. Convolutional Neural networks (CNN) is also applied to this task. One of the major challenges of learning-based methods is collecting and annotating a great amount of sun glare perturbed data from diverse scenes.

To tackle this problem, in this paper, we first propose a sun glare image generation method based on the physical model and Generative Adversarial Network (GAN). The physical model first produces the sun glare pattern based on ray tracing through the optical system to the film plane. Next, a glare-affected image is generated by adding a glare pattern image to a glare-free driving scene image. Finally, the glare-affected image is further refined by a CycleGAN [3] model for a more realistic effect.

In order to ensure the safety of the perception system, we propose an offline robustness benchmark to the sun glare for model performance validation and a robust online sun glare detection method for safety monitoring with the help of our generated glare-affected images. In the offline robustness benchmark, different severity levels of sun glare are injected and an evaluation measure is also defined so as to assess the robustness of object detection models. For the online glare detector, we train a CNN-based segmentation model in a self-supervised manner. The training data are generated during training following multiple random factors like glare pattern, position, dimension, etc., to avoid overfitting problems. With experiments on both real and synthetic sun glare data, our detection model shows a superior performance compared to the state-of-the-art methods.

To summarize, the main contributions of this paper are followings:

1. A glare-affected generation method by combining physical model and GAN model
2. A sun glare robustness benchmark for performance assessment of object detection models.
3. A sun glare detection approach trained in a self-supervised way superior to the state-of-the-art performance.

2 Related work

Glare generation. One class of glare generation method [4] is the use of static textures such as starbursts, circles, and rings that move according to the position of the light source and are composited additively to the base image. These methods are simple and efficient, but they fail to capture natural lens flare’s intricate dynamics and variations. And the other class of methods [5] involves ray or path tracing through a virtual lens with all of its optical elements. These approaches, in general, deliver a high rendering quality. But it is very costly to compute. Instead of considering individual rays, Hullin et al. [6] exploited the strong correlation of rays within a light bundle to accelerate the rendering process.

Glare detection. There are only a few previous existing works on glare or over-exposure detection on images. The authors in [7, 8, 9] used the brightness value of the pixels as a glare feature and a threshold was applied to extract the glare-affected or overexposed regions in the image. Besides the brightness, Guo et al. [10] also uses color based features for detection. Singh et al. [11] combined local thresholding, intensity based thresholding and color channel thresholding to determine the glare regions in night scene images. Yahiaoui et al. [12] applied a threshold to get only pixels that

are potentially over-exposed. To remove noise, a closing operation has been applied followed by an erosion operation which removes small objects so that only substantive objects remain. Then they perform a blob detection by finding the contours. However, all these image post processing methods do not work well without adjusting the threshold parameter to adapt the glare detection on different scenes. And other bright regions on the image like bright sky, white cars can easily lead to false positive glare detections. Andalibi et al. [13] proposed to combine the shape features and the thresholding on contrast and saturation map. The Hough transform was applied following the assumption that the glare region is round. This feature helps to reduce false-positive detections. But it cannot detect the reflected glare on the road, which is not in round shape.

Deep learning methods are also applied to glare detection in some previous works. Yahiaoui et al. [12] created a new sun glare dataset and used this dataset to train CNN for detection. They further proposed to fuse the result of a thresholding based image processing method and the CNN. Rodin et al. [14] proposed to use a CNN to detect glare area in document images. The main problem of these methods is the limited number of data used for training (580 glare images in [12], 692 glare images in [14]). The annotated data is not sufficient for training a deep neural network. This introduces the bias in training a convolution network. The training set covers only a small part of real-world glare-affected scenes, making it impossible to guarantee the model’s performance when tested on different image sources.

Some existing work leveraged the localization information to help the glare detection task. In [15], the authors proposed a method to use the Google Street View (GSV) panorama images to predict the occurrence of sun glare. They used a segmentation model to classify those obstruction pixels from GSV images. The occurrence of sun glares is determined by sun’s angle in relation to cars and obstructions. Aune et al. [16] presented a method to determine the presence of sun glare on the road at any time by calculating the angle and position of the sun and taking the surrounding terrain into account, including prediction of sunrise and sunset time. Andalibi et al. [17] extracted feature maps using lightness, saturation, contrast, and color distance and then refined them based on the sun’s predicted location from the GPS information. Similarly, these methods cannot detect the reflected glare.

Instead of analyzing the scene, another type of approach focused on detecting dazzling effect on the driver’s face. For example, in [18] the difference between the averaged pixel value of the face region and the selected ceiling part is monitored. This class of method can be used in Advanced Driver Assistance Systems (ADAS), but cannot be used in autonomous systems.

In this work, we propose to combine the glare generation method and the deep learning model to perform glare detection. The CNN can extract more robust features than the simple brightness feature. Moreover, glare image generation allows self-supervised training without data collection and annotation specific to sun glare scenes. More diverse data used in training procedures leads to better detection performance and penalization ability.

Benchmarking corruption robustness. Several existing works investigated the influence of image corruption on CNN models. Doge et al. [19] evaluated different state-of-the-art image classification CNNs under different types of distortions like blur, Gaussian noise. Azulay et al. [20] quantified the lack of invariance of several modern CNNs to small translations. A benchmark to evaluate the robustness of recognition models [21] and object detection models [22] against common corruptions was recently introduced. In this work, we study the vulnerability of CNN detector to the sun glare corruptions. To the best of our knowledge, robustness benchmark to sun glare is not investigated yet in the state of the art.

3 Method

An overview of the proposed method is shown in Fig 2. In this section, we first summarize the physically-based glare rendering method as preliminary. Then we present our glare-affected image generation method by a pattern image overlaying and GAN refinement. In the end, we introduce our offline robustness benchmark and the glare detection method based on synthetic glare images.

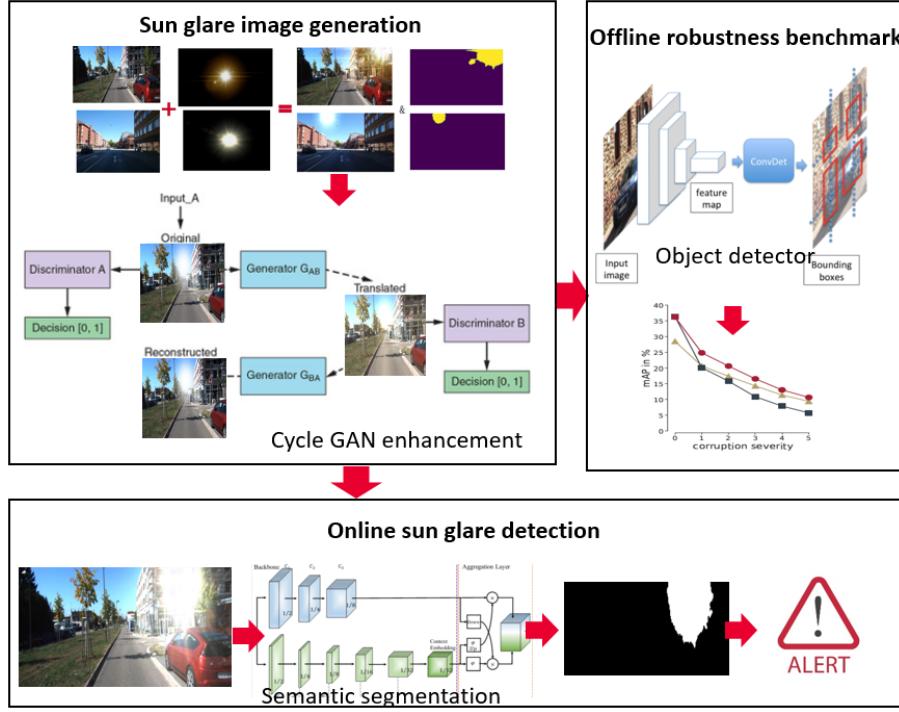


Figure 2: The overview of our proposed method. Glares are generated and rendered by physical modal. GAN is used for refinement. The synthetic glare images are further used for self-supervised detection training and offline robustness assessment.

3.1 Glare pattern generation

We follow [6] to generate the glare patterns. Light passing through a transmissive aperture and propagating further through free space is diffracted into a circular cross-section. This diffraction causes the typical starburst centered on the light source at the aperture. Such shapes can be computed using the Fourier transform of the aperture function. Adding some noise to the aperture function can increase the realism of the effect.

Every time a ray of light hits an interface between two media, part of the light is reflected at optical surfaces. Given n optical surfaces, there are $\frac{n(n-1)}{2}$ possible second-order interreflections. Each corresponds to a flare element called ghost. A sparse bundle of rays is traced through the lens and mapped to a grid in the sensor plane for each ghosting artifact. The ghosts are then rendered in the rasterization unit. The final shape of the ghost is then obtained by combining the corresponding textures. By shooting rays for different wavelengths, chromatic effects can be reproduced naturally. The starburst and ghosting artifacts are added up as the final glare rendering result. By varying optical system model parameters like the wavelength of light and angle of incidence, we can produce different glare effects. In order to diversify the glare patterns, we finally form two glare datasets. One consists of the generated glares, and the other consists of the glares captured by real lens from [23] (see Fig 3). More details will be given in section 4.1.1.

3.2 Synthesize glare-affected images

Sun glare only appears under specific time and weather conditions. It's hard to collect a large amount of sun glare affected images in different scenes. To deal with this problem, we propose to synthesize the glare-affected images, which allows us to obtain glare-affected image under various scenes by combining glare pattern and clear images.

In order to further add the pattern diversity, we augment the generated and captured glare pattern by applying random geometric and color transformations. For geometric augmentation, we perform random flip as well as random rotation, scale and aspect ratio within a certain range. For color



Figure 3: Sun glare pattern examples. (a) Generated glare patterns. (b) Captured natural sun glare

augmentation, we change randomly the color temperature and brightness of the pattern image. The glare pattern is padded to the dimension of glare-free image and is centered at a random position on that image. The border padding of refraction mode is applied to reduce visual artifacts. We model a glare as a semitransparent layer on top of the underlying scene. A glare-affected image is synthesized by adding a padded glare pattern to a glare-free natural image in linear space. Color channels are only clipped after combining the augmented glare with the clean image.

3.3 GAN based refinement

The addictive over-layer introduces a local glare effect. In order to add a more realistic effect and make the global style of the synthetic image closer to the natural glare affected images, we use the Cycle-GAN modal introduced by Zhu et al. [3] to perform an unpaired image-to-image translation for refinement.

There are two generators (G_A and G_B) and two discriminators (D_A and D_B) for the respective image translation tasks in the Cycle-GAN setup. The task of the generator G_A is to take an image from class B as an input and generate an image which appears to be from class A. In our case, we take the synthetic image as class A and the real glare-affected image as class B. The task of the discriminator D_A is to distinguish images of class B from the input dataset and the images generated by the generator. The model setup is to have a minimax game between the discriminators and generators.

In addition to the usual GAN loss, a cycle loss constrains the generated images to be similar to the input images. There is also an optional identity loss between input and output of generators to prevent the generator from introducing wild color changes and artifacts. The final refined image examples are shown in Fig. 4.



Figure 4: Synthetic glare-affected image examples.

3.4 Self-supervised sun glare detector

Although sun glare has firm characteristics, the real-world glare presents vari color temperature, shape, and under-layer behind semitransparent halo. As a consequence, the classical image processing method does not work well on glare detection. Hence, We leverage the strength of CNN for feature extraction. We use a binary segmentation CNN to detect and localize the glare-affected area.

The purpose of the glare detection task is for automated vehicles to be self-aware of perception uncertainty. Therefore, the computational resource used for this monitoring function should be limited. So we choose Bisenet-v2 [24] (based on Mobilenet-v2 [25]) architecture as the CNN backbone, which is designed for efficient, low consuming inference. In Bisenet-v2, low-level and high-resolution detail features and high-level semantic context features are aggregated in a lightweight structure to achieve a good trade-off between speed and accuracy. Pixel-wise cross-entropy loss is applied to train the network.



Figure 5: Online generated training image and label examples.

The glare effect presents a soft boundary. This makes the annotation work not evident, and hard to unify an annotation standard. In our case, the ground truth can be obtained by thresholding the brightness on the padded sun glare image, which allows us to train the CNN in a self-supervised way without manually annotating. So we propose to generate online training examples and labels following section 3.2 with data augmentation (see Fig. 5). Since performing the GAN refinement online is computationally costly. We randomly refine some training samples. Combining clear driving scene images and various glare patterns can make a considerable amount of training data, and no duplicate image appears during training. This helps to avoid overfitting and increases the generalization ability of the model. Moreover, the data collection and labelling work are fully reduced by this self-supervised training procedure.

3.5 Robustness benchmark to sun glare

We introduce the robustness benchmark to sun glare inspired by the PASCAL-C benchmark [22] assessing object detection robustness on corrupted images. Similar to [22], in order to assess the vulnerability, we introduce several severity levels of sun glare corruption. From the color temperature change caused by simulated glare, glare standing at the border of the bounding box, to glare invading the bounding box and partially covering the object (see Fig 6). We evaluate the performance of the object detector for different severity levels. More details will be given in section 4.2.



Figure 6: The glare severity levels from 0 to 5 with respect to a specific vehicle object (the vehicle with a bounding box).

4 Experiments

In section 4.1, we perform experiments to evaluate the effectiveness of our glare detection model. In section 4.2 we show the robustness benchmark results.

4.1 Glare detection experiments

4.1.1 Dataset

We form two datasets for the glare detection evaluation.

Synthetic sun glare data: We use the Kitti dataset [26], one of the most commonly used datasets for driving scene object detection. We follow the instructions in section 3.2 and 3.3 to add glare effect to the test set of Kitti. Since the glare is generated with the ground truth mask, we use the intersection of Union (IoU) to evaluate the glare detection performance. As to the glare patterns used for synthesizing, we use 20 collected real glare patterns, 40 synthetic glare for training and another 20 real glare for testing.

Real sun glare data: The dataset used in [12, 17] are not publicly available yet. We collected the natural glare-affected images either from web sources or by our camera installed on vehicle. In total, we obtain 6k images with sun glares (see Fig. 7). One half is used for the Cycle-GAN training, and the other half is used as the detection test set. Since there is no pixel-level annotation on the real glare images. We add another 3k driving scene images without sun glare to form the test set with as many positive samples as negative samples. We perform an image-level evaluation, i.e., we classify whether an image contains glare. The detection precision, recall, accuracy are used as evaluation measures.



Figure 7: Real sun glare-affected image examples from our test set.

4.1.2 Experimental setup

We set up different variants of our methods for comparison. For the baseline, we generated offline 1k sun glare-affected images. And we train the CNN detector in supervised mode. This is the analogy of the method trained on the limited annotation data in [12]. We also train another three glare detection CNNs in self-supervised mode with respectively 1k images from the Kitti training set, 7k images from the Kitti training set, and 30k images by mixing up images in the Kitti, cityscapes [27], BDD100k [28] and nuScenes [29] datasets. The input of the CNN is 640×416 . Adam optimizer is used for training. The learning rate is set to 0.002. Besides the sun glare rendering, only random flip and crop are performed as data augmentation. We train for 300, 100 and 30 epochs respectively for the settings with 1k, 7k and 30k training data.

4.1.3 Glare detection results

The final evaluation results are shown in Table. 1, and some detection examples on real data are shown in Fig. 9. Whether on the real or synthetic test set, our method outperforms the existing state-of-the-art methods widely. It shows that it achieves the best result with the 30k training data and the more data used in glare image synthesizing, the better the performance is. This shows the effectiveness of our proposed self-supervised learning.

The image processing methods got a lower IoU on synthetic setting evaluation due to large blocks of false detection on skies got a lower recall on real glare settings evaluation because of misdetection for warm but not such bright glares. The baseline and the self-supervised detector with the 1k training data got close results on synthetic settings since both over-fitted to the Kitti scenes. In real glare settings, the self-supervised detector outperforms the baseline. This proves the validity of our proposed self-supervised method again.

Table 1: The glare detection results on our synthetic and real datasets.

	Synthetic glare dataset		Real glare dataset		
	Glare IoU	Bg IoU	Precision	Recall	Accuracy
Overexposure [8]	16.9	87.5	86.7	29.7	66.4
Saturation&Contrast([17] w/o GPS)	24.6	87.7	61.1	79.4	68.0
Baseline, similar to [12]	73.5	95.7	57.0	97.4	65.8
Ours with kitti-1k	74.2	96.2	64.8	97.5	75.1
Ours with kitti-7k	83.5	98.0	69.4	91.3	78.0
Ours with mixed-30k	88.7	98.7	98.7	91.1	95.5

4.2 Robustness benchmark to sun glare

We use YOLO-v3 [30] as the baseline detector of the benchmark. YOLO-v3 is a one-stage detection method that directly converts the positioning problem of the object bounding box into a regression problem for processing. We train two different detectors with different glare augmentation settings, i.e., with or without a glare rendering during training. The rendering is performed randomly with a probability ratio of 0.2. And we control that the glare mask occults each object bounding boxes by less than 50% area. We follow common conventions to select the training and validation splits (80/20). We randomly pick one single object on each image in the validation set. And glares are introduced with respect to the selected object at a specific severity level. Since the glare causes only false negatives, we use the detection rate for the affected objects rather than the mean Average Precision (mAP) as the benchmark metric.

The result is shown in Fig. 8. It shows that the YOLO-v3 detector suffers seriously from glare corruption, and adding glare rendering augmentation helps yielding a better performance under glare corruptions. This benchmark can be an indicator to determine the sensibility of an online glare alert.

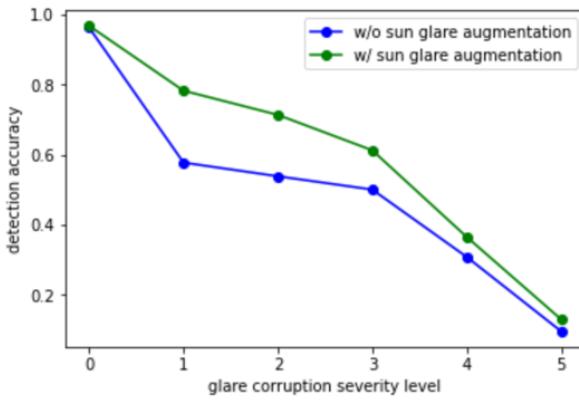


Figure 8: The glare robustness benchmark results.

5 Conclusion

In this paper, in order to achieve online and offline autonomous driving self-awareness for sun glare corruption, we introduced a sun glare detection approach and robustness benchmark to sun glare corruption based on glare rendering. The glare is generated by a bundle tracing method and refined by a Cycle-GAN. Moreover, we make use of glare synthesizing to train a glare detector in a self-supervised manner. With the experiments on both synthetic and natural datasets, our approach shows superior results to state-of-the art methods. Then we create a robustness benchmark to assess the performance reliability of objection detectors. We implement a baseline detector and show that the detection performance can be notably affected by sun glare.

References

- [1] Donald A Redelmeier and Sheharyar Raza. Life-threatening motor vehicle crashes in bright sunlight. *Medicine*, 96(1), 2017.
- [2] Kenji Hagita and Kenji Mori. Analysis of the influence of sun glare on bicycle or pedestrian accidents with turning vehicle. In *13th World Conference on Transport Research*, 2013.
- [3] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [4] MJ Kilgard. Fast opengl-rendering of lens flares. 2000.
- [5] J Chaumond. Realistic camera-lens flares. 2007.

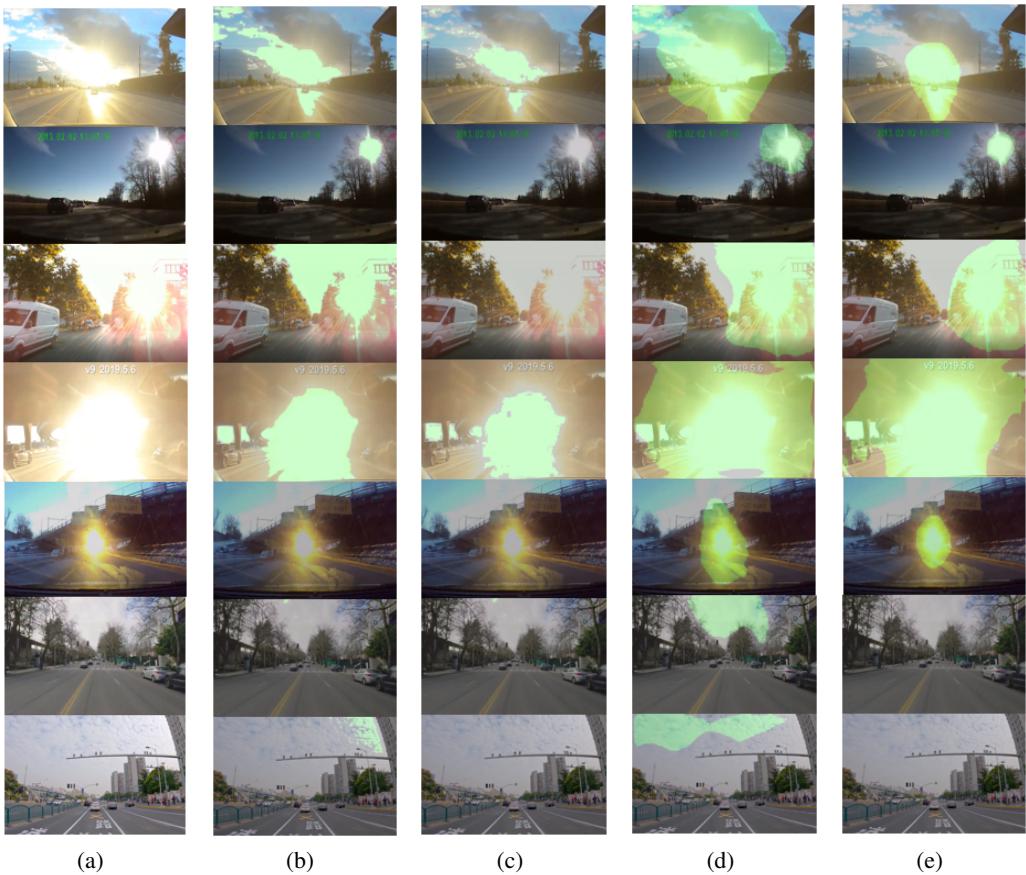


Figure 9: Sun glare detection result examples. (a) Original images. (b) Overexposure [8]. (c) Saturation & Contrast([17] w/o GPS). (d) Baseline, similar to [12]. (e) Ours with mixed-30k.

- [6] Matthias Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. Physically-based real-time lens flare rendering. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011.
- [7] Suchart Pharadornpanitchakul and Roungsan Chaisricharoen. Danger detection from head light glare through vision estimation. In *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, pages 661–664. IEEE, 2013.
- [8] Dae-Hong Lee, Yeo-Jin Yoon, Seok-jae Kang, and Sung-Jea Ko. Correction of the overexposed region in digital color image. *IEEE Transactions on Consumer Electronics*, 60(2):173–178, 2014.
- [9] Likun Hou, Hui Ji, and Zuowei Shen. Recovering over-/underexposed regions in photographs. *SIAM journal on imaging sciences*, 6(4):2213–2235, 2013.
- [10] Dong Guo, Yuan Cheng, Shaojie Zhuo, and Terence Sim. Correcting over-exposure in photographs. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 515–521. IEEE, 2010.
- [11] Mandakinee Singh, Rajesh Kumar Tiwari, Kunal Swami, and Ajay Vijayvargiya. Detection of glare in night photography. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 865–870. IEEE, 2016.
- [12] Lucie Yahiaoui, Michal Uřičář, Arindam Das, and Senthil Yogamani. Let the sunshine in: Sun glare detection on automotive surround-view cameras. *Electronic Imaging*, 2020(16):80–1, 2020.
- [13] Mehran Andalibi and Damon M Chandler. Automatic glare detection via photometric, geometric, and global positioning information. *Electronic Imaging*, 2017(19):77–82, 2017.
- [14] Dmitry Rodin and Nikita Orlov. Fast glare detection in document images. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 7, pages 6–9. IEEE, 2019.

- [15] Xiaojiang Li, Bill Yang Cai, Waishan Qiu, Jinhua Zhao, and Carlo Ratti. A novel method for predicting and mapping the occurrence of sun glare using google street view. *Transportation research part C: emerging technologies*, 106:132–144, 2019.
- [16] Kjetil Aune. Sun glare detection and visualization. Master’s thesis, [K. Aune], 2017.
- [17] Mehran Andalibi, Kousuke Kawai, and Damon M Chandler. An algorithm for glare detection via photometric, colorimetric, and global positioning features. *Artif. Intell. Res.*, 9(1):12–26, 2020.
- [18] Xindong Liu, Adrian Leu, D Bacara, M Hainfellner, and Axel Graeser. Robust dazzling detection in a novel visual based dazzling avoidance system. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 833–838. IEEE, 2014.
- [19] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2016.
- [20] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.
- [21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.
- [22] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.
- [23] shutterstock.com.
- [24] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, pages 1–18, 2021.
- [25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [27] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [28] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.
- [29] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [30] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.