

Manav Prabhakar, Jwalandhar Girnar, Arpan Kusari*

University of Michigan Transportation Research Institute
2901 Baxter Road, Room 202,
Ann Arbor, MI-48103
{prmanav, jwala, kusari}@umich.edu

Abstract

While much research has recently focused on generating physics-based adversarial samples, a critical yet often overlooked category originates from physical failures within on-board cameras—components essential to the perception systems of autonomous vehicles. Camera failures, whether due to external stresses causing hardware breakdown or internal component faults, can directly jeopardize the safety and reliability of autonomous driving systems. Firstly, we motivate the study using two separate real-world experiments to showcase that indeed glass failures would cause the detection based neural network models to fail. Secondly, we develop a simulation-based study using the physical process of the glass breakage to create perturbed scenarios, representing a realistic class of physics-based adversarial samples. Using a finite element model (FEM)-based approach, we generate surface cracks on the camera image by applying a stress field defined by particles within a triangular mesh. Lastly, we use physically-based rendering (PBR) techniques to provide realistic visualizations of these physically plausible fractures. To assess the safety implications, we apply the simulated broken glass effects as image filters to two autonomous driving datasets—KITTI and BDD100K—as well as the large-scale image detection dataset MS-COCO. We then evaluate detection failure rates for critical object classes using CNN-based object detection models (YOLOv8 and Faster R-CNN) and a transformer-based architecture with Pyramid Vision Transformers. To further investigate the distributional impact of these visual distortions, we compute the Kullback-Leibler (KL) divergence between three distinct data distributions, applying various broken glass filters to a custom dataset (captured through a cracked windshield), as well as the KITTI and Kaggle cats and dogs datasets. The KL divergence analysis suggests that these broken glass filters do not introduce significant distributional shifts. Our goal is to provide a robust, physics-based methodology for generating adversarial samples that reflect real-world camera failures, with the overarching aim of improving the resilience and safety of autonomous driving systems against such physical threats.

Introduction

Cameras are ubiquitous as remote sensors, collecting data from an unstructured and dynamic external environment, often in harsh conditions. A failure or fault in a sensor is a divergence from the functional state in at least one given parameter of the system (van Schrick 1997). These faults can occur due to internal (such as wear and tear) or external (temperature, humidity etc) causes. For RGB cameras, internal causes include dead pixels while external causes include fractured enclosures or outer lens, and condensation. These abrupt failures are hard to detect and negatively impact object detection algorithms - reducing accuracy and often leading to hallucination as shown in Fig. 1. The failures occurring in an automated vehicle (AV) for example, can lead to critical safety issues resulting in crashes and in some cases, fatalities.

Currently, to the best of authors' knowledge, there are no rigorous methods for generating camera based sensor failures (Ceccarelli and Secci 2022).

In this work, we focus on the sensor failure occurring due to fractures in any glass covering a camera (or camera enclosure), although the process detailed in this paper can be used for any of the camera failures listed in (Ceccarelli and Secci 2022). These glass fracture effects in a camera can be caused due to an external object hitting the camera or as a result of heat and/or pressure developing suddenly within the enclosure. In the parlance of neural networks, an image captured in such conditions is considered as an adversarial sample. Previous research (Akhtar and Mian 2018; Carlini and Wagner 2017; Szegedy et al. 2013) shows that even small amounts of corruptions, sometimes difficult to be seen by human eyes, are enough to completely fool the neural networks where a subtle change of inputs can lead to a drastic change in outputs. We would like to note that (Li, Schmidt, and Kolter 2019) provided a physical camera-based adversarial attack paradigm, which serves as the closest related work in this domain. They presented a modification of the image using an overlay of a translucent, carefully crafted sticker which led to misclassification.

To understand the effect of these fractures on the resulting camera images, we conducted two distinct experiments: one

Code —
<https://github.com/manavprabhakar/camera-failure>

*Corresponding author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Կոտրված ապակի, անհաջող տեսախցիկները Ֆիզիկայի վրա հիմնված հակառակորդական սմուշների սիմոլյացիա ինքնավար վարորդական համակարգերի համար

Manav Prabhakar, Jwalandhar Girnar, Arpan Kusari*¹ Միջիզանի
Համապարփակ տրանսպորտային հետազոտությունների ինստիտուտ 2901
Baxter Road, Սթելլա 202, Էն Արքր, MI-48103 {prmanav, jwala, kusari}
@umich.edu

arXiv:2405.15033v3 [cs.CV] 14 Nov 2025

మీ — <https://github.com/manavprabhakar/camera-failure>

Ամփոփում

Ներածություն
Տեսախցիկները ամենուր են որպես հեռավոր տեսարողներ, հավաքելով տվյալներ չշառուցված և դինամիկ արտաքին միջավայրից, հաճախ կրող պայմաններում: Սենսորի խափանումը կամ սխալ համակազիք առնվազն մեկ տրված պարամետրի ֆունկցիոնա վիճակից շեղում է (van Schricker 1997): Այս սխալները կարող են առաջանալ ներքին (օրինակ՝ ամշվածություն) կամ արտաքին (գերմաստիճան, խոնավություն և այլն) պատճառներով: RGB տեսախցիկների համար ներքին պատճառները ներառում են մեռած պիզետներ, իսկ արտաքին պատճառները՝ կուրորված պատյաններ կամ արտաքին ոսանյակներ և խտացում: Այս հանկարծակի խափանումները դժվար է հայտնաբերել և բացասարա են ազդում օբյեկտի հայտնաբերման ագրորիթմների վրա՝ նվազեցնելով ճշգրտությունը և հաճախ հանձեցնելով հայրությանը, ինչպես ցույց է տրված նկար 1-ում: Օրինակ՝ ավտոմատացված տրանսպորտային միջոցում (ԱՎ) տեսի ունեցող խափանումները կարող են հանգեցնել անվտանգության լուրջ խնդիրների, որոնք կարող են հանգեցնել վթարների և որոշ դեպքերում՝ մահվան:

Ներկայումս, բայց հետինակների լավագույն գփտեիրների, չկան խիստ մերողներ տեսախցիկների վրա հմմաված սենառների ձախողումների գեներացման համար (Ceccarelli և Secci 2022):

Այս աշխատանքում մենք կենտրոնանում ենք սենսորների խափանումների վրա, որոնք առաջանում են տեսախցիկի (կամ տեսախցիկի պատյանի) ապակու ճաքերի պատճառով, թեև այս հողածում մանրամասնված գործընթացը կարող է օգտագործել (Ceccato & Secci 2012) նշյալ տեսախցիկի խափանումներից ցանկացածի համար: Տեսախցիկի պատյան ճաքերի ազդեցությունները կարող են առաջանալ արտաքին օրյեկտի հարվածից կամ պատյանի ներսում հանկարծակի զարգացող ժերմության և/կամ ճնշման հետևանքով: Նեյրոնային ցանցերի լեզվով՝ նման պայմաններում նկարահանված պատկերը համարվում է հակառակորդական նմուշ՝ նախորդ հետազոտություններ (Akhtar և Mian 2018; Carlini և Wagner 2017; Szegedy և այլ 2013) ցոյց են տալիս, որ նոյնիվ փոքր քանակությամբ արակարությունները, որոնք երեսն ոժվար է տեսնել մարդկային աշքով, բավկան են նեյրոնային ցանցերը ամբողջությամբ խաբելու համար, որտեղ մուտքերի փոքր փոփոխությունը կարող է հանգեցնել եթերի կտրուկ փոփոխության: Մենք ցանկանում ենք նշել, որ (Li, Schmidt և Kolter 2019) ներկայացրել են ֆիզիկական տեսախցիկի վրա հիմնաված հակառակորդական հանգակման արարագիմ, որը ծախայում է որպես այս ոլորտում ամենասուր հարակից աշխատանք: Նրանք ներկայացրել են պատկերի փոփոխություն՝ օգտագործելով կիսաքաղաքանացիկ, խնամքով մշակված կարոնի ծածկույթ, որը հանգեցրել է սխալ դասակարգման:

Այս ճեղքերի ազդեցությունը տեսախցիկի ստացված պատկերների վրա հասկանալու համար մենք իրականացրեցինք երկու տարբեր փորձարկումներ՝ մեկը

*Համապատասխանը հեղինակ Եղիշևակյան Իրավունք © 2026,
Արենտական բանականության առաջնադաշտման
ասոցիացիան (www.aaai.org): Բոլոր հոգածները պատճենավոր են:

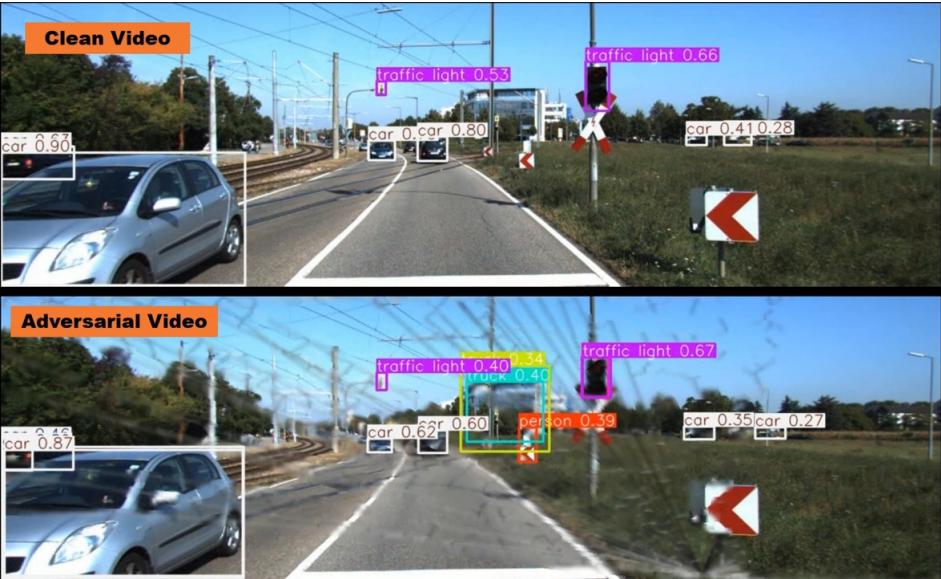
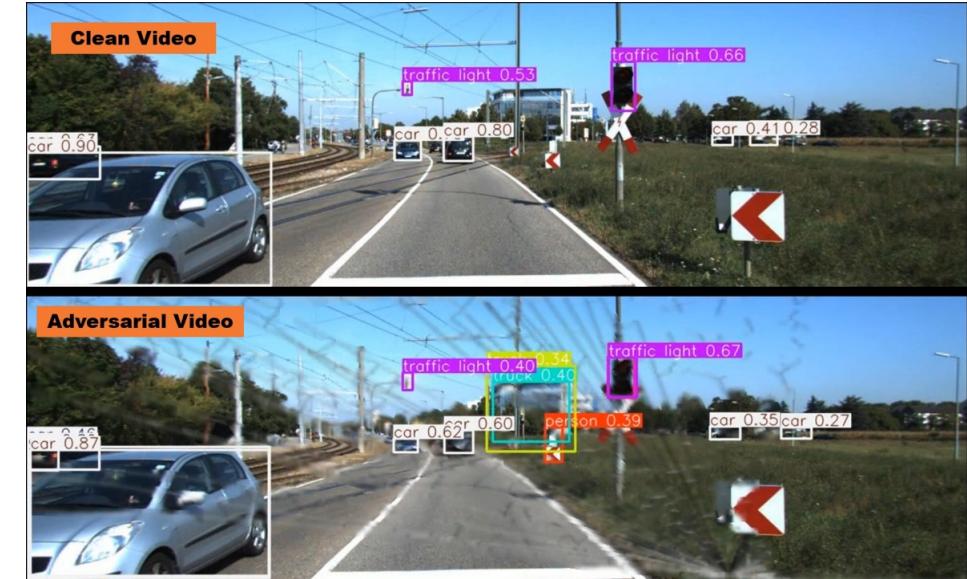


Figure 1: A qualitative comparison of clean vs adversarial video generated using our simulation and rendering method on KITTI. This frame shows false positives, and reduced confidence levels for true positives. Refer to the supplementary material for full video.

in an indoor static environment and the other in a dynamic outdoor environment. The first one involved fracturing tempered glass and placing it in front of the camera (see Fig. 2(a)) with a static vehicle in the scene to understand how different fracture patterns affect the quality and appearance of the scene. We captured images at different focal lengths to judge the variability of such corruptions. This helped us answer certain qualitative questions about the visual appearance of these fractures with respect to their spread and intensity, motivating our approach in Section Focal Plane and Physical attack simulation. The experimental setup and the detailed experimental results are in Sec. Static Experiment of Supplementary. The second experiment (Fig. 2(b)) consisted of recording an outdoor video with dynamic vehicles under daylight conditions by placing a MobileEye camera next to a windshield crack presented in Fig. 2 (shown in the upper left) and performing inference using YOLOv8 (Jocher, Chaurasia, and Qiu 2023) to gain a primitive understanding of the impact of such scenarios on object detection networks. We observed that the model can easily detect the vehicle in a clean image while it suffers from detection failure (lower right) or generates false positives (lower left). Interestingly, the presence of a crack can also unexpectedly increase the confidence in prediction of the car presenting a clearly defined edge (0.92 in the lower left vs. 0.75 in the upper left). The detailed inference results with vehicle and person class is given in Sec. Dynamic Experiment of Supplementary.

We then looked for real broken glass images online (Sec. Real glass fracture images of Supplementary) but failed to build a dataset large enough to enable a data-driven approach for adversarial defense for these conditions. Additionally, we experimented with CGI tools like Maya and Blender for



Նկար 1: Սաքրու և հակառակորդային տեսանյութերի որակական համեմատություն, որոնք ստեղծվել են մեր սիմուլացիայի և պատկերման մեթոդի միջոցով KITTI-ի վրա: Այս կադրը ցույց է տալիս կեղծ դրականներ և վստահության մակարդակի նվազում դրական դրականների համար: Լրիվ տեսանյութի համար դիմեր հավելվածին:

creating such effects but they lack the flexibility, control, scale and physics to simulate these conditions. The closest simulation option in existing literature is ArcSim (Pfaff et al. 2014). However, their high-resolution simulation outputs are extremely slow (≈ 20 hours), making it difficult to scale. As a result, we directed our efforts towards creating a scalable simulation-based pipeline for generating fractures that can be used to advance perception stack.

For a glass fracture, the principal point, force and angle of incidence may be random, but the spread and the resulting pattern follows an inherently physical process (being either linear or radial). We thus build a fracture simulation based on particles in a triangular mesh generated randomly and perform stress propagation through the mesh. Our simulation allows us to produce the fractures within a triangular mesh at every discrete time state δt . We use OpenCV to convert the given mesh to a corresponding broken glass pattern image. We then utilize physically-based rendering (PBR) (Pharr, Jakob, and Humphreys 2023) to realistically render the surface fractures using bidirectional reflectance distribution function (BRDF) by calculating the amount of light reflected from a given point on a surface as a result of source(s) of light being incident on it.

Combining our rendering approach with three popular open source datasets - KITTI (Geiger et al. 2013), BDD100k (Yu et al. 2020) and MS-COCO (Lin et al. 2014), we are able to generate adversarial images efficiently. A common process for testing the generated adversarial images is to find the number of false positives/negatives across the image space. However, in our case, due to the adversarial effect being local, we cannot rely simply on an image based measure. We therefore, use the adversarial images (similar to the lower left figure of Fig. 2) and extract the objects

ներսի ստատիկ միջավայրում և մյուսը դինամիկ արտաքին միջավայրում: Առաջինը ներառում էր տեմպերացված ապակու ճեղքումը և այն տեսախցիկի առջև տեղադրելը (տես նկ. 2(a)): Տեսարանի մեջ ստապումեքնայի հետ, որպեսզի հասկանանք, թե ինչպես են տարբեր ճեղքաձիգ նախշերը աղդում տեսարանի որակի և տեսքի վրա: Մերժ նկարահանել ենք պատկերներ տարբեր ֆուլա երկարություններով՝ գնահատելու նման աղավաղումների փոփոխականությունը: Սա մեզ օգնեց պատասխանել որոշ որական հարցերի այլ ճեղքերի տեսողական տեսքի վերաբերյալ՝ կապված դրանց տարածման և հիմնային դիմումների հետ, ինչը խթանեց մեր մուտեցումը Ֆուլա հարաբերական հարաբերական սիմուլացիա բաժնում: Փորձարկման կարգավորումը և մանրամասն փորձարկման արդյունքները տրված են Համեմատված Ստատիկ փորձարկում բաժնում: Երրորդ փորձարկում (նկ. 2(b)) ներառում էր դինամիկ ավտոմումեքնայի հետ բացօթյա տեսանյութի նկարահանումը ցերեկային պայմաններով՝ MobileEye տեսախցիկը տեղադրելով առջևի ապակու ճեղքի կողքին, որը ներկայացված է նկ. 2-n ցուցարկած է վերին ձախում) և YOLOv8 (Jocher, Chaurasia, and Qiu 2023) օգտագործելով կանխատեսում կատարելուն մասնաւոր սենարների ազդեցությունը օբյեկտի հայտնաբերման անցեցի վրա նախական հասկանայի համար: Մենք նկատեցինք, որ մոտենած հետությամբ կարող է հայտնաբերել ավտոմումեքնան մարդու պատկերում, մինչդեռ այն տառապան է հայտնաբերման ձախողում (ներքի աջում) կամ ստեղծում է կեղծ դրականներ (ներքի ձախում): Հետաքրքի է, որ ճեղքի ավելացույնը կարող է նաև անսպասիլիորեն բարձրացնել ավտոմումեքնայի կանխատեսում վստահությունը՝ ներկայացնենք հստակ սահմանված էղոր (0.92 ներքին ձախում ընդունեմ 0.75 վերին ձախում): Մարդամասն կանխատեսման արդյունքները ավտոմումեքնայի և անձի դաշտ հետ տրված են Համեմատված դինամիկ փորձարկում բաժնում:

Այսուհետև մենք փորձեցինք գտնել իրական կոտրված ապակու պատկերներ առցանց (Համեմատված իրական ապակու կոտրվածի պատկերներ բաժնում), բայց չկարողացանք ստեղծել բավարար մեծ տվյալների հավաքածու, որու թույլ կտար տվյալների վրա հիմնված մուտեցում մշակել այս պայմանների համար հակառակորդային պաշտպանության համար: Բայց այդ, մենք փորձարկեցինք CGI գործիքներ, ինչպիսիք են Maya-ն և Blender-ը:

Այսպիսի էֆեկտներ ստեղծելու համար, բայց դրանք չունեն նկույտություն, վերահսկողություն, մասշտար և ֆիզիկա՝ այս պայմանները մոնթեվիդու համար: Գյուղայուն ունեցող գրավանության մեջ ամենամու մոնթեվիդու հարթեակ ԱրցՍիմ-ն (Pfaff et al. 2014): Սակայն, Կա ան բարձր լուծաչափում մոնթեվիդու արդյունքները չափազանց դանդաղ են (≈ 20 ժամ), ինչը դժվարացնում է մասշտարավորումը: Արդյունքում, մենք մեր ցանքերը ուղղություն կոտրվածներ ասեղծուու մասշտարացնում մոնթեվիդու վրա հիմնված կոտրվաչափա ստեղծելու ուղղությունը: Որ կարող է օգտագործել նկալման համակարգի գործույթը:

Ապակու կոտրվածի դեպքում հիմնական կետը, ուժը և անկունը կարող են ինեւ պատահական, բայց տարածումը և արդյունքուն առաջացող նախյուն են ընդու ֆիզիկական գործնական հիմնով գծային կամ ճառագայթային: Ուստի մենք կառուցմ ներ կոտրվածի սիմուլացիա, որը հիմնված է պատահականութեա գեներացված առանձին ցանքություններու վրա կատարում է առաջական բարձրացնում գանցի մասնիկների վրա և կատարում ենք յարկածության տարածում ցանքի միջոցով: Մեր փորձարկում թույլ է տալիս մեզ ստեղծել կոտրվածներ եռանկյուն ցանքուն ներում յուրաքանչյուր դիսկրետ ժամանակային վիճակում: Մենք օգտագործում ենք OpenCV տրված ցանքը համապատասխան լուրդված ապակու նախյա պատկերը համար: Այսուհետև մենք օգտագործում ենք ֆիզիկական հիմնված պատկերում (PBR) (Pharr, Jakob, and Humphreys 2023): Իրականական պատկերներն ապակուն մասերն ապակու կոտրվածները՝ օգտագործույն երկողունակ անդրադարձան բաշխման ֆոնկցիան (BRDF) հաշվարկերը մասերն ապակու կոտրվածները տրված կետից անդրադարձան պատկերը արդյունքը:

Մեր պատկերներ ստեղծման մուտեցում համարելով երե հայտնի բաց կորպ տվյաների հավաքածուների հետ՝ KITTI (Geiger et al. 2013), BDD100K (Yu et al. 2020) և MS-COCO (Lin et al. 2014), մենք կարող ենք արդյունավետուն ստեղծել հակառակորդային պատկերներ: Ստեղծված հակառակորդային պատկերների փորձարկման ընդհանուր գործնական պատկերային տարածում վիճակն արդյունքների քանակի որոշումն է: Սակայն, մեր դեպքում, հակառակորդային աղեցույթունը ինեւ տեղային, մենք չենք կարող պարզացան ապակուներ պատկերային շահման վրա: Ուստի, մենք օգտագործում ենք հակառակորդային պատկերները (նման նկ. 2-ի ներքի ձախում): Հետո մեր պատկերներին կուտար մասերը կատարում է աղեցույթը:

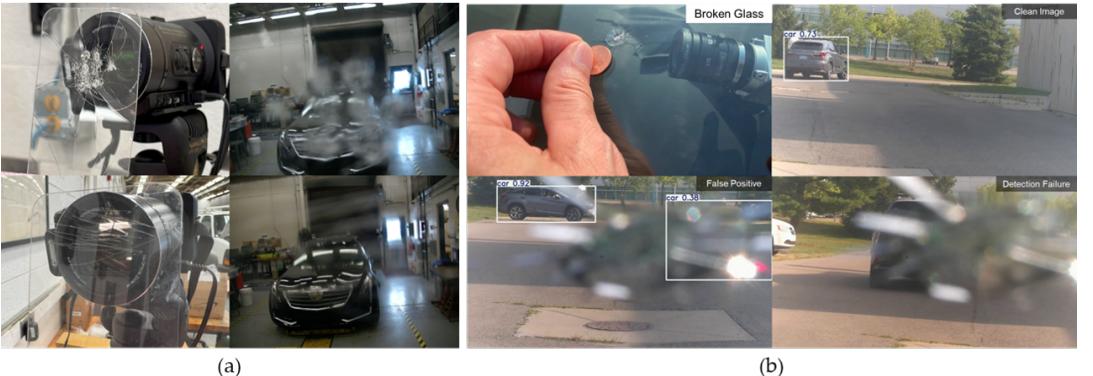


Figure 2: (a) Indoor static experiment. Left: Camera with 2 different fractured tempered glass patterns; right - images of the vehicle under the different fractures. (b) Outdoor dynamic experiment. Top left - a coin sized windshield crack; top right - clean image with the vehicle detected using YOLOv8; bottom left - false positive through the crack; bottom right - detection failure through the glass. More examples from these experiments have been provided in the supplementary material.

which lie within the region where the fracture exists using the ground truth bounding boxes. We then utilize YOLOv8, Faster R-CNN (Ren et al. 2016) and Pyramid Vision Transformer (PVTv2) (Wang et al. 2022) to find the percentage of objects that fail when the adversarial filters are applied. We also provide ablation studies to understand the distributional differences between the three set of images: Real broken glass images collected experimentally, real broken glass images collected online and the generated images. We compute the Kullbeck-Liebler (K-L) divergence for these image distributions to prove similarity of the generated images to the real broken glass images. We utilize cat images from Kaggle Cats and Dogs dataset as control to understand the difference between image distributions (PK).

The major contributions of the paper can be summarized as follows:

$$W_{\alpha} = \frac{1}{2} \left(1 - \frac{1}{\sqrt{1 + \frac{4}{3} \sin^2(\alpha)}} \right) \approx \frac{1}{2} \left(1 - \frac{1}{\sqrt{1 + \frac{4}{3} \sin^2(0.01)}} \right) = 0.4999$$

- We provide a novel way of abstracting glass fracture through a combination of stress propagation methods and minimum spanning trees, to generate physically sound broken glass patterns.
 - We present a PBR approach to facilitate a realistic render of camera failures that can be used with any kind of existing computer vision datasets - both images and videos.
 - Our simulation and rendering pipelines are scalable and computationally efficient ($\approx 1.6s$) allowing it to be used by both academia and industry for enhancing robust and out of distribution protection for a wide range of applications.

Background

Physics based adversarial samples

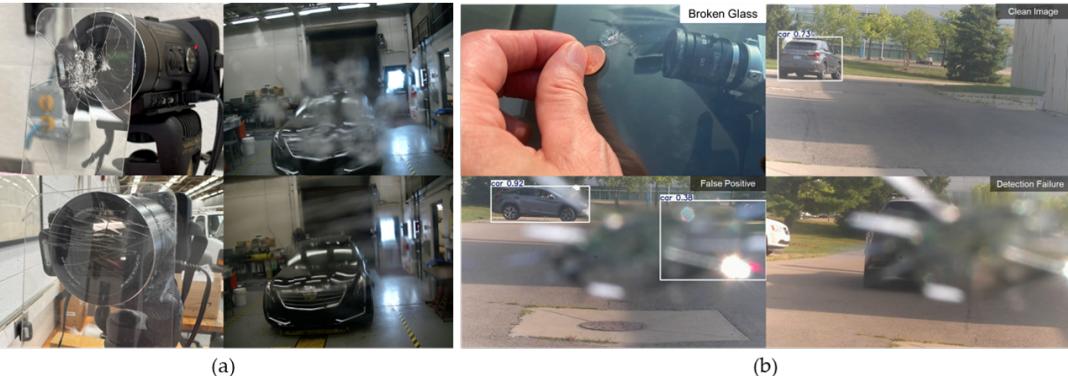
The problem of adversarial sample can be defined as follows: for a model M that classifies an input sample X correctly to its designated class i.e. $M(X) = y_{true}$, adding an error ϵ to the input sample X , results in an altered sample \hat{X} such that $M(\hat{X}) \neq y_{true}$. Thus, the injection of the error ϵ results in an adversarial sample that causes the model to fail.

Although the idea of adversarial manipulation of the model has been identified in the context of machine learning quite some time ago (Dalvi et al. 2004), in the last decade, the focus has squarely been on the adversarial attacks on neural networks (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014). In these papers, the researchers showed that a small targeted injection of noise, almost imperceptible to the human eye, changed the labels completely (Szegedy et al. 2013) and conversely, images could be generated that looked completely unrecognizable to humans but which had perfect classifications from the DNNs (Nguyen, Yosinski, and Clune 2015).

While these adversarial samples probe the model for possible failures, they lack any physical realism behind their generation and need access to the model. To address this, some recent research has targeted building physically relevant adversarial samples. One of the first forays into this was made by (Kurakin, Goodfellow, and Bengio 2018) who targeted the accuracy of the models in the physical world by feeding noisy images from a cell-phone camera that led the model to incorrectly classify a large fraction of the samples. Along the similar vein, (Eykholt et al. 2018) demonstrated that real traffic signs can be perturbed with simple physical stickers placed strategically to fool state-of-the-art DL algorithms almost perfectly even with viewpoint changes. Other researchers have placed adversarial images (Kong et al. 2020), translucent patches on camera (Zolfi et al. 2021) or artificial LiDAR surfaces (Tu et al. 2020) to generate samples which fool object detectors. While these prior research use physics in terms of generating the samples, they do not come from modeling a rigorous physical process and we aim to fill this gap in this work.

Cracked/fractured glass theory

The subject of how glass breaks and how it propagates is still an open research question and one that has been contentious with multiple physical theories being proposed (Roussel and Brow 2012). While the microscopic procedure of glass crack is being debated on, on a macroscopic level, the cracking



Նկար 2: (ա) Ներսի ստատիկ փորձարկում: Ձևի կողմում՝ տեսախցիկը 2 տարրեր կուրոված կարծ ապակու նախշերով; աջ կողմում՝ տրանսպորտային միջոցի պատկերները տարրեր կուրովածների տակ; (բ) Դրսի հիմանմի փորձարկում: Վերևի ձախ կողմում՝ մետաղադրամի չափի ճարպած դիմացակու վրա; Վերևի աջ կողմում՝ մաքր պատկեր, որտեղ տրանսպորտային միջոցը հայտնաբերված է ՅՈԼՕՎ-ի միջոցով; Ներքի ձախ կողմում՝ միայլ դրամական հայտնաբերում ճարպի միջոցով: Ներքի աջ կողմում՝ հայտնաբերման ձախողում ապակու միջոցով: Այս փորձարկումների ավելի շատ օրինակներ ներկայացված են հավելվածում:

Հոդվածի հիմնական ներդրումները կարելի է ամփոփել հետևյալ կերպ.

- Մենք առաջարկում ենք ապակու կոտրվածքի վերացալանացման նոր եղանակ՝ սրբեսի տարածման մեթոդների և նվազագույն ծածկող ծառերի համարդրությամբ՝ ֆիզիկապես հիմնավորված ապակու նախշեր ստեղծելու համար:
 - Մենք ներկայացնում ենք PBR մուտեցում՝ տեսախցիկի խափանումների իրատեսական պատկերով հետուացնելու համար, որը կարող է օգտագործելի ցանկացած գործորուն ունեցող համակարգային տեսողության տվյալների հավաքածուների հետ՝ իշխան պատկերների, այնպես էլ տեսանալություն:
 - Մեր սիմուլացիոն և աստղերային գեներացման խողովակաշարերը մասշտաբային են և հաշվարկային արդյունավետ ($\approx 1.6s$), ինչը թույլ է տալիս այն օգտագործել ինչ պես ակադեմիայի, այնպես էլ արդյունաբերության կողմից՝ դիմացկանության և բաշխումից դրու պաշտպանության բարելավման համար՝ լայն կիրառությունների շրջանակում:

Հիմնավորում

ֆիզիկայի վրա հիմնված հակառակորդային նմուշներ

Հակառակորդի նմուշի խնդիրը կարելի է սահմանել հետևյալ կերպ. մոդելի համար M , որը ճիշտ դասակարգում է մուտքային նմուշը X իր նշանակված դասին, այսինքն՝ $M(X) = \text{true}$, մուտքային նմուշին ϵ սխալի՝ ավելացնելով հանգեցնում է փոխիրարկ նմուշի X' , այնպես որ $M(X) \neq \text{true}$. Այսպիսով, սխալ ϵ ներակումը հանգեցնում է հակառակորդի նմուշի, որը մոդելի ձախողման պատճառ է դառնում:

Թեև մոդելի հակառակորդային մանխպայտացիայի գաղափարը մերենայական ուսուցման համատեքստում հայտնաբերվել է բավականին վաղուց (Dalvi et al. 2004), վերջին տասնամյակում ուշադրությունը կենտրոնացել է ենթռնային ցանցերի վրա հակառակորդային հարձակումների վրա (Szegedy և այլ 2013; Goodfellow, Shlens, և Szegedy 2014): Այս հոդվածներում հետազոտողները ցուց տվեցին, որ փոքր թիրախավորված աղոմվի ներարկումը, որը գրեթե անկատեի է մարդկային աշքի համար, ամբողջությամբ փիսում է պիտուղները (Szegedy և այլ 2013), և ընդհակառակը, կատարի է ստեղծել պատկերներ, որոնք մարդկանց համար լիովին անճնանաշեի են, բայց որոնք ունեն կատարայի դասակարգումներ DNN-ներից (Nguyen, Yosinski, և Clune 2015):

Չնայած այս հակառակորդային նմուշները մողեղ ստուգով են հանրավիր ճախբուղմների համար, դրանց ստեղծման հետևող չկա ֆիզիկական իրատասություն, և անհրաժեշտ է մողեղի հասանելիություն։ Այս խնդիրը լուծելու համար վերջին որոց հետազոտությունները ուղղված են եղել ֆիզիկապես համապատասխան հակառակորդային նմուշների ստեղծմանը։ Առաջին փորձերից մեւը կատարվել է (Kurakin, Goodfellow, և Bengio 2018) կորմից, ովքեր նախասար էին դրե՛ մողեղների ճշգրտությունը ֆիզիկական աշխարհում՝ բջջային հեռախոսի տեսախցիկոց առմկուն ապահովեներ մատակարարելով, ինչը մողեղին ստիպում էր սիսալ հասակարգել նմուշների մեծ մասը։ Նմանատիպ ուղղությամբ, (Eyxholo և այլը 2018) ցոյց տվեցին, որ իրավան ճանապարհային նշանները կարող են խանգարվել պարզ ֆիզիկական պիտօններով, որոնք ուղանավարկանը դեռևս կատարված են՝ գրեթե կատարայ եերապու խարելու ժամանակակից DL ալգորիթմներին, նովինիկ տեսանկյունին փոփոխություններով։ Այլ հետազոտողներ տեսարդի են հակառակորդային պատկերներ (Kong և այլը 2019), թափանցիկ կարկասաններ տեսախցիկի վրա (Zoell և այլը 2021) կամ արհեստական LiDAR մակերեսներ (Tun և այլը 2020)¹ նմուշներ համար, որոնք խարեւում են օբյեկտների դետեկտորներին։ Չնայած այս նախորդ հետազոտությունները ֆիզիկան օգտագործում են նմուշների ստեղծման համար, դրանց չեն գալիս իհաստ ֆիզիկական գործընթացի մողեղավորությից, և մենք նպատակ ունենք լրացնել այս բաց այս այլասարքություն։

Ծաքածկության պահանջման առաջնային գործություններ

Թեման, թե ինչպես է ապակին կոտրվում և ինչպես է այն տարածվում, դեռևս բաց հետազոտական հարց է, և այն եղել է վիճակարուց՝ առաջարկված բազմաթիվ ֆիզիկական տեսություններով (Rouxel և Brow 2012): Մինչ ապակու ճարի միկրոսկոպիկ գործընթացը քննարկվում է, մակրոսկոպիկ մակարդակում ճարերի



Figure 4: (a) Shows the simulated image with the road and vehicles in the focal plane (PBR and Far-focus). (b) denotes the simulated crack pattern in the focal plane (PBR and short focus).

Table 1: Average precision (in percentage) of different classes in KITTI, BDD100k and MS-COCO under different adversarial images. x provides the overlay relation between dataset and glass-crack type. Clean x Dataset - refers to directly the particular images without any adversarial sample. RO x Dataset - refers to Real images of cracked glass collected online overlayed on clean images. Sim x Dataset - refers to simulated crack patterns overlayed on clean images.

Dataset	IoU threshold	Category	Clean x Dataset	RO x Dataset	Sim x Dataset
KITTI (YOLOv8)	0.5	Pedestrian	25.64	69.72	17.84
		Truck	12.39	3.59	3.76
		Car	58.99	50.7	57.73
	0.75	Pedestrian	6.83	33.88	6.02
		Truck	11.29	2.67	2.79
		Car	31.25	23.85	30.15
BDD100k (PVTv2)	0.5	Pedestrian	66.47	54.33	25.95
		Truck	61.97	52.83	52.02
		Car	80.37	70.14	56.78
	0.75	Pedestrian	27.06	22.72	10.60
		Truck	47.03	38.23	42.52
		Car	46.23	45.97	42.99
MS- COCO (Faster R-CNN)	0.5	Person	0.035	0.024	0.024
		Vehicles	2.14	1.45	1.87
		Food	35.34	28.07	30.65
	0.75	Person	0.032	0.022	0.023
		Vehicles	1.56	1.05	1.07
		Food	24.59	18.85	22.00

the trend is maintained with the pedestrian class showing the steepest drop. For MS-COCO, we aggregated the AP for the super - categories: person, vehicles and food. This is because a lot of objects in MS-COCO occupy smaller area in the image frame making it difficult to get meaningful results from all categories. A very intriguing result is that the pedestrian class has a multifold increase in AP under the real broken glass patterns. While this trend might seem counter-intuitive, it resonates with the results in Fig. 2 where the confidence of the car increases because of an edge. This in fact shows that the AP is highly dependent on the crack pattern making it extremely important to create defense methodologies to mitigate these adversarial attacks.

Ablation studies

Our results indicate that the simulated images obtain a similar adversarial effect as the real images. Thus, an important ablation study for us is to understand how close the simulated crack patterns are to the real cracked glass patterns and those collected online. We form 5 distributions

- Crack patterns collected online (Fig. 5 top left)
- Simulated crack patterns (Fig. 5 bottom left)
- Simulated crack patterns overlayed on KITTI (Fig. 5 bottom right)
- Crack patterns collected online overlayed on KITTI (Fig. 5 top right)

We now compute the K-L divergence among all these distributions to compute how similar they are to each other (see Fig. 6). In order to provide a control, we compare KITTI to images of cats from the Kaggle dataset, providing a K-L divergence of 2.434. In that scale, the PBR images of broken glass have a difference of 0.36 to the real broken glass patterns while the broken glass filters overlaid on KITTI images have similar K-L divergence.

Fig. 7 shows an analysis of the computation time for each of our modules and over different number of particles. We perform this analysis on 100 runs, generating random impact points, impact angles, and mesh structure with a fixed number of particles. The difference in computation time for different runs can be attributed to the impact point and im-



Նկար 4: (ա) ցույց է տալիս սիմոլացված պատկերի ճանապարհով և տրանսպորտային միջոցներով ֆոկալ հարթությունում (PBR և հեռավոր ֆոկուս): (բ) նշում է սիմոլացված ճաքերի նախշը ֆոկալ հարթությունում (PBR և կարծ ֆոկուս):

Աղյուսակ 1: Միջին ճշգրտություն (տոկոսներով) տարբեր դասերի համար KITTI, BDD100k և MS-COCO-ում տարբեր հակառակորդային պատկերների ներք: x-ը ապահովում է տվյալների հավաքածուի և ապակու ճաքերի տեսակի միջև հարաբերությունը: Clean x Dataset - վերաբերում է անմիջապես տվյալ պատկերներին առանց որևէ հակառակորդային նմուշի: RO x Dataset - վերաբերում է առցանց հավաքված ճաքակու իրական պատկերներին, որոնք դրված են մաքրու պատկերների վրա: Sim x Dataset - վերաբերում է սիմոլացված ճաքերի նախշերին, որոնք դրված են մաքրու պատկերների վրա:

Տվյալների հավաքածու	IoU շեմ	Կատեգորիա	Clean x Dataset	RO x Dataset	Sim x Dataset
KITTI (YOLOv8)	0.5	Հետիոտն	25.64	69.72	17.84
		Բնօնատար	12.39	3.59	3.76
		Car	58.99	50.7	57.73
BDD100k (PVTv2)	0.75	Հետիոտն	6.83	33.88	6.02
		Բնօնատար	11.29	2.67	2.79
		Car	31.25	23.85	30.15
MS- COCO (Faster R-CNN)	0.5	Հետիոտն	66.47	54.33	25.95
		Բնօնատար	61.97	52.83	52.02
		Car	80.37	70.14	56.78
	0.75	Հետիոտն	27.06	22.72	10.60
		Բնօնատար	47.03	38.23	42.52
		Car	46.23	45.97	42.99
MS- COCO (Faster R-CNN)	0.5	Մարդ	0.035	0.024	0.024
		Հրամանադրություն միջոցներ	2.14	1.45	1.87
		Food	35.34	28.07	30.65
	0.75	Մարդ	0.032	0.022	0.023
		Հրամանադրություն միջոցներ	1.56	1.05	1.07
		Food	24.59	18.85	22.00

տենդենց պահպանվում է, և հետիոտնի դասը ցույց է տալիս ամենախստ անկումը: MS-COCO-ի համար մենք համախմբեցիք AP-ն սուպեր-կատեգրիաների համար՝ ան, տրանսպորտային միջոցներ և սնուն: Սա այն պատճառով է, որ MS-COCO-ում շատ օբյեկտներ գրանցվում են ավելի փոքր տարածք պատկերի շրջանակում, ինչը դվյանցում է բոլոր կատեգրիաների հմաստալից արդյունքներ ստանալիք: Չափ հետաքրի արդյունք է, որ հետիոտնի դասը բազմապատճի աճ է ցույց տալիս AP-ում իրավան կոտրված ապակու նախշերի տակ: Թեև այս տենդենցը կարող է հակառակն թվայի այն համարուն չ նկ. 2-ի արդյունքներին, որում ավտոմեքենայի վսանաւությունը մեծանում է եղան պատճառով: Սա իրավան նույն է տալիս, որ AP-ն իսկս կախված է ճաք նախշից, ինչը շափական կարող է դարձնում պաշտպանական մերժաբանությունների ստեղծման՝ այս հակառակորդային հարձակումները մեղմելու համար:

Աբայցին ուսումնադրություններ

Մեր արդյունքները ցույց են տալիս, որ սիմոլացված պատկերները ստանում են ննանատիպ հակառակ ազդեցություն, ինչպես իր ական պատկերները: Հետևաբար, մեզ համար մենք համեմառն ենք KITTI-ն Kaggle տվյալների հավաքածուի կատունների պատկերների ներ ստանալով 2.434 K-L չեղում: Այդ սանդղակում, կոտրված ապակու PBR պատկերները 0.36 տարբերություն ունեն իրավան կոտրված ապակու նախշերից, մինչդեռ KITTI պատկերների վրա դրված կոտրված ապակու ֆիլտրերը ունեն նման K-L չեղում:

Նկ.7-ը ցույց է տալիս մեր մոդուլների յուրաքանչյուրի հաշվարկման ժամանակի վերլուծությունը տարբեր քանակի մասնիկների համար: Մենք այս վերլուծությունը կատարում ենք 100 փորձակումների վրա՝ գներուաներով պատճառական ազդեցության կետեր, ազդեցության անկյուններ և ցանցի կառուցվածք՝ ֆիլտրված մասնիկների քանակությունը: Տարբեր փորձարկումների հաշվարկման ժամանակի տարբերությունը կարող է վերագրվել ազդեցության կետերին և ան-

- Real on-road dataset (depicted in Fig. 2)

- Առցանց հավաքված ճաքերի նախշերը (Նկ. 5 վերևի ձախ)
- Մոդելավորված ճաքերի նախշեր (Նկ. 5 ներքևի ձախ)
- Մոդելավորված ճաքերի նախշեր, որոնք դրված են KITTI-ի վրա (Նկ. 5 ներքևի աջ)
- Ցանցում հավաքված ճաքերի նախշերը դրված են KITTI-ի վրա (Նկ. 5 վերևի աջ)

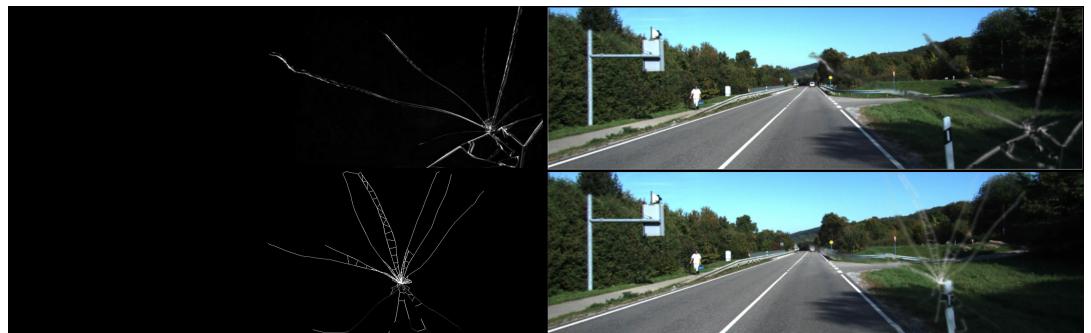


Figure 5: Top left - Crack pattern collected online on Freepik; top right - online crack pattern overlayed on KITTI; bottom left - simulated crack pattern with PBR; bottom right - simulated crack pattern overlayed on KITTI.

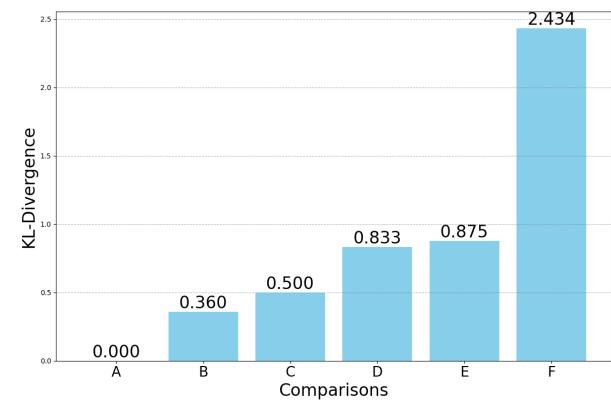


Figure 6: K-L divergence of different pairs of image distributions. Datasets: RC - Real on-road dataset (see Fig. 2), KITTI and Cats. Filters: RO - Real (collected online) and Sim - Simulated. K-L divergence between (x - overlay relation): A - (Sim x KITTI) vs (Sim x KITTI); B - (Sim vs RO); C - (Clean RC vs KITTI); D - (Broken RC) vs (RO x KITTI); E - (Broken RC) vs (Sim x KITTI); F - KITTI vs Cats.

pact angle. The cracking visualization and render time also vary owing to different sized masks formed due to varying fracture patterns. We also vary the number of particles and see how runtime increases exponentially with the increase in particles. All these runs were rendered on images from the KITTI dataset with dimensions of $(375 \times 1242 \times 3)$.

Conclusion and Future Scope

We have introduced a novel class of adversarial failures resulting from the physical process of failures in the camera. In this paper, we provide an approach to generate a realistic broken glass pattern from a physical simulation and subsequently embed that to existing image datasets using physically based rendering. We show that the simulated adversarial images can lead to significant errors in object detection.

In this work, we address black-box adversarial attacks stemming from real-world, naturally occurring physical phenomena, not artificially crafted to exploit specific model

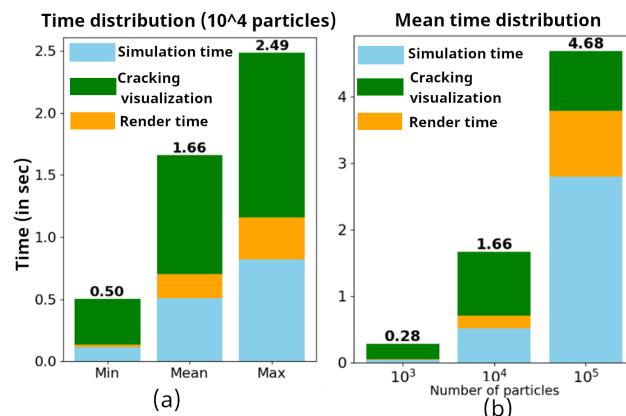
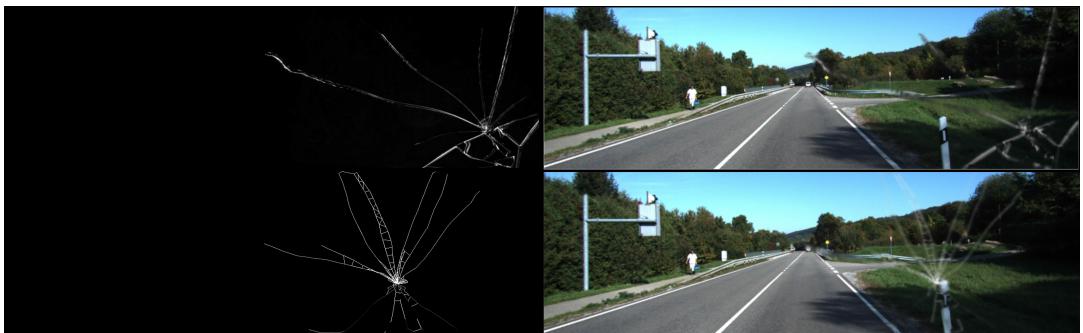


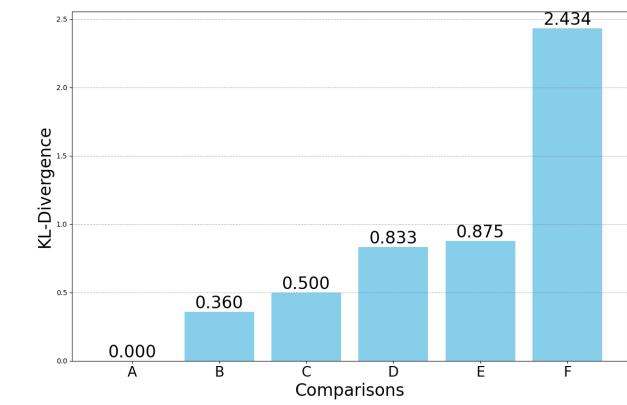
Figure 7: (a) Mean time taken by different modules of our pipeline across 100 runs. (b) The minimum, maximum and mean time taken by different modules across 100 runs for a 10^4 particle mesh. For these plots, we showcase the time taken for simulation (simulation time), converting the mesh to glass (cracking visualization) and finally rendering (render time).

vulnerabilities. We assume no knowledge of the model attributes, weights or architecture, ensuring attacks are transferable across various models. Physical adversarial methods (Translucent Patch, RP2) can all be termed as occlusions of either the camera or the objects being captured. The adversariality comes from the effect of the model inference due to these occlusions. Our PBR pipeline blends the cracks with source images as translucent, blurry patterns, impacting latent space encoding rather than causing direct occlusion, resulting in incorrect detections.

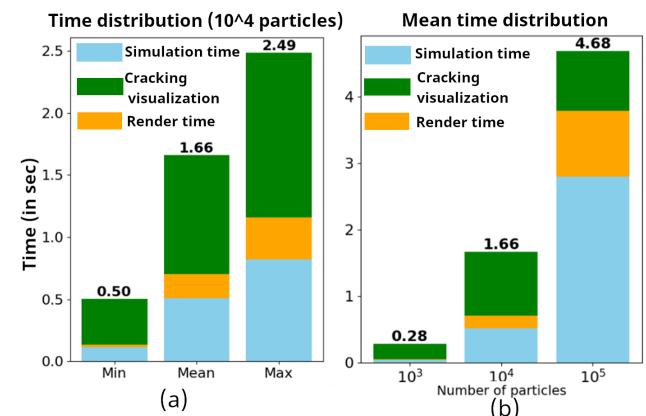
While this work introduces a physics-based method for broken glass pattern generation specifically, camera failures encompass other effects such as sun-glare, overexposure, underexposure, condensation etc. Our future work will focus on creating an adversarial toolbox for realistic generation of these effects using physics and subsequently, placing them on existing image datasets and car simulation platforms to promote further research in this field of partial camera failures.



Նկ. 5: Վերևի ձախ - Ցանցի օրինակ, հավաքված առցանց FreePik-ում; Վերևի աջ - առցանց ցանցի օրինակ, դրվագ KITTI-ի վրա; Ներքևի ձախ - PBR-ով մոդելավորված ցանցի օրինակ; Ներքևի աջ - մոդելավորված ցանցի օրինակ, դրվագ KITTI-ի վրա:



Նկ. 6: K-L շեղումը տարբեր գովազ պատկերների բաշխումների միջև:
 Տվյալաշարքը: RC - Իրական ճանապարհային տվյալաշարք (տես Նկ. 2), KITTI և Կատուններ: Զտիշներ: RO - Իրական հավաքած առցանց) և Sim - Սրբելավորված: K-L շեղումը (x - համադրության հարաբերություն): A - (Sim x KITTI) vs (Sim x KITTI); B - (Sim vs RO); C - (Մարոր RC vs KITTI); D - (Կոտորված RC) vs (RO x KITTI); E - (Կոտորված RC) vs (Sim x KITTI); F - KITTI vs Կատուններ:



Նկար 7: (ա) Մեր խողովակաշարի տարրեր մոդուլների կողմից 100 վագրի ընթացքում միջին ժամանակը: (բ) Տարրեր մոդուլների կողմից 100 վագրի ընթացքում նվազագույն, առավելագույն և միջին ժամանակը 10^4 մասնիչային ցանցի համար: Այս գծապատճեններում մնանք ցուցաբերում ենք սիմուլացիայի համար պահանջվող ժամանակը (սիմուլացիայի ժամանակը), ցանցը ապակու վերածելու գործընթացը (նաքերի տեսանելիություն) և վերօնական ապատկերում (ապատկերման ժամանակը):

Եզրակացություն և ապագա տեսլական

Մենք Ներկայացրել ենք հակառակորդային խափանումների նոր դաս, որոնք առաջանում են տեսախցիկի խափանումների ֆիզիկական գործընթացից: Այս հոդվածում մենք առաջարկում ենք մոտեցում՝ ֆիզիկական մողելավորված ստացածքի հրական կուրորիա ապակու օրինակ ստեղծելու և այն ֆիզիկական հիմքով պատկերներ տվյալաշրջերին ներդնելու համար: Մենք ցույց ենք տալիս, որ մողելավորված հակառակորդային պատկերները կարող են հանգեցնել օբյեկտի հայտնաբերման էական փսխաների:

Այս աշխատանքում ենք անդրադառնում ենք ան արկի հակառակորդային հարձակումներին, որոնք ծագուի են իրական աշխատանքի, բնականորեն առաջացող ֆիզիկական երևություններից, ոչ թե արհեստականորեն ստեղծված՝ հատուկ մոդելի թերությունները շահագործելու հմար:

իսոցելիություններ: Մենք ենքարդում ենք, որ մողեկի հատկությունների, քաշերի կամ ճարտարապետության մասին չի մի գիտելիք չունենք, ապահովելով, որ հարձակումները փոխանցելի են տարբեր մողեկների միջև: Ֆիզիկական հակառակորդային մեթոդները (Քափանցիկ կարկասան, RPZ) կարող են բրոյր համարվել որպես հոչընդոտներ կամ տեսախցիկի, կամ նկարահանվող օբյեկտների: Հ ակառակորդայնությունը զայն է մողեկի եղանակացության ազդեցությունից՝ այս հոչընդոտների պատճառով: Մեր PBR խողովակաշարը խառնվում է ենթերը սկզբնադրյալ պատկերների հետ՝ որպես քանակական, մշշոյն նախշեր, ապելու քանված տարածության կողմանական վրա՝ փոխանցեն ուղղակի հոչընդոտման, ինչը հանգեցնում է միասնականությունների:

Թեև այս աշխատանքը ներկայացնում է ֆիզիկայի վրա հմտնված մեթօդ՝ կուրոված ապակու նախշերի գեներացման համար, տեսահսկի խափանումները ներառում են նաև այլ ազդեցություններ, ինչպիսիք են արևի շողը, գերլուսավորումը, թրենուսավորումը, խտացումը և այլն։ Մեր ապագան աշխատանքը կլենորոնան այս ազդեցությունների հրականականացնելու համար հակառակորդային գործիքազի ստեղծման վրա՝ օգտագործելով ֆիզիկա, և այնուհետև դրանք տեղադրելով առկա պատկերների տվյալների հավաքածուների և ավտոմեքենաների սիմուլացիոն հարթակների վրա։ Հետագա հետազոտությունները մասնակի տեսահսկի խափանումների որոշ լուրջ։

References

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zolfi, A.; Kravchik, M.; Elovici, Y.; and Shabtai, A. 2021. The translucent patch: A physical and universal attack on object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15232–15241.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; и Darrell, T. 2020. BDD100K. Տարերակված վարորդական տվյալների հավաքածու բազմաբնույթ ուսուցման համար: *IEEE/CVF* համաժողով համակարգչային տեսողության և նախշերի ճանաչման վերաբերյալ

2636–2645: Zolfi, A.; Kravchik, M.; Elovici, Y.; и Shabtai, A. 2021. Թափանցիկ կարկատան. ֆիզիկական և համընդհանուր հարձակում օբյեկտի դետեկտորների վրա: *IEEE/CVF* համաժողով համակարգչային տեսողության և նախշերի ճանաչման վերաբերյալ

15232–15241:

Algorithm of stress propagation

Algorithm 1 describes the procedure for simulating the propagation of stress through a material following an impact event. The algorithm takes as inputs the location of the impact (pt), the magnitude of the impact force (F), the impact direction vector (v), and the parent edge (PE) associated with the impact site. It also uses a nearest neighbor radius R to determine the set of candidate locations for stress propagation.

Algorithm 1: Stress Propagation

```

1:  $pt \leftarrow$  Impact Point
2:  $F \leftarrow$  Impact Force
3:  $PE \leftarrow$  Parent Edge
4:  $v \leftarrow$  Impact Vector
5:  $R \leftarrow$  Nearest neighbor radius
6:
7: procedure PROPAGATESTRESS( $Pt, F, V, PE$ )
8:    $frontiers \leftarrow KDT tree - queryRadius(R)$ 
9:    $NN \leftarrow \frac{frontiers - pt}{\|frontiers - pt\|}$ 
10:   $\cos(\theta) \leftarrow NN \cdot v$ 
11:   $stress \leftarrow calculateStress(\cos(\theta), F)$ 
12:   $frontiers \leftarrow frontiers[argmax(stress)]$ 
13:   $v \leftarrow v[argmax[stress]]$ 
14:   $PE \leftarrow PE[argmax[stress]]$ 
15:  PROPAGATESTRESS( $Pt, F, V, PE$ )
16: end procedure

```

First, it uses a KD-tree data structure to efficiently query all points ($frontiers$) within a given radius R of the impact point. For each frontier, it computes a unit direction vector from the impact point to the frontier (NN). It then projects the impact vector v onto this direction to obtain the cosine similarity $\cos(\theta)$, capturing the angular relationship between the impact direction and the candidate propagation direction. For each candidate, the resulting value is used, together with the impact force, to calculate the corresponding stress at that point. The algorithm then selects the candidate with the maximum stress value. The impact vector v and parent edge PE are updated to correspond to this new direction. The process is recursively repeated, allowing the simulated stress wave to propagate iteratively through the material along the path of greatest stress transfer.

This approach aims to mimic how stress from an impact point is most likely to radiate through a material—preferentially following paths defined by both geometric proximity and mechanical alignment with the original impact.

The final output of the simulation is the realization of the mesh as an image which corresponds to broken lens pattern (final image of Fig. 8).

Հարվածության տարածման ալգորիթմ

Ալգորիթմ 1-ը նկարագրում է նյութի միջով լարվածության տարածման սիմուլացիայի ընթացակարգը հարվածից հետո: Ալգորիթմը դրվել մուտքային տվյալներ ընդունում է հարվածի տեղը (pt), հարվածի ուժի մեծությունը (F), հարվածի ուղղության վեկտորը (v) և հարվածի վայրին կապված ծնողական եզրը (PE): Այն նաև օգտագործում է մուտքային հարվածի շառավիղը R' լարվածության տարածման թեկնածու վայրերի հավաքածուն որոշելու համար:

Ալգորիթմ 1: Լարվածության տարածում

```

1:  $pt \leftarrow$  Հարվածի կետ
2:  $F \leftarrow$  Հարվածի ուժ
3:  $PE \leftarrow$  Ծնողական եզր
4:  $v \leftarrow$  Հարվածի վեկտոր
5:  $R \leftarrow$  Ամենամոտ հարվածի շառավիղ
6:
7: գործընթաց PROPAGATESTRESS( $Pt, F, V, PE$ )
8:    $frontiers \leftarrow KDT tree - queryRadius(R)$ 
9:    $NN \leftarrow \frac{frontiers - pt}{\|frontiers - pt\|}$ 
10:   $\cos(\theta) \leftarrow NN \cdot v$ 
11:   $stress \leftarrow calculateStress(\cos(\theta), F)$ 
12:   $frontiers \leftarrow frontiers[argmax(stress)]$ 
13:   $v \leftarrow v[argmax[stress]]$ 
14:   $PE \leftarrow PE[argmax[stress]]$ 
15:  PROPAGATESTRESS( $Pt, F, V, PE$ )
16: ավարտել
ընթացակարգը

```

Սկզբում այն օգտագործում է ԿԴ-ծառ տվյալների կառուցվածքը՝ արդյունավետորեն հարցման համար բոլոր կետերը (սահմանները) տրված շառավիղով R ազդման կետից: Յուրաքանչյուր սահմանագծի համար այն հաշվարկում է միավոր ուղղության վեկտոր ազդման կենդանագծ (NN): Այնուհետև այն նախագծում է ազդման վեկտորը v այս ուղղության վրա՝ ստանալու կոմիուսային նմանությունը $\cos(\theta)$, որը դրսում է անվյունային հարաբերությունը ազդման ուղղության և թեկնածու տարածման ուղղության միջև: Յուրաքանչյուր թեկնածուի համար ստացված արժեքը օգտագործվում է ազդման ուժի հետ միասին՝ այդ կետում համապատասխան լարվածությունը հաշվարկելու համար: Ալգորիթմը այնուհետև ընտրում է առավելագույն լարվածության արժեքը ունեցող թեկնածուին: Ազդման վեկտորը v և ծնողական եզրը PE թարմացվում են՝ համապատասխանելու այս նոր ուղղությանը: Գործընթացը կրկնվում է ուղղակի կերպով, թույլ տալով, որ մոդելավորված լարվածության այլքը տարածվի նյութի միջով առավելագույն լարվածության փոխանցման ուղիղով:

Այս մոտեցումը նպատակ ունի նմանակել, թե ինչպես է լարվածությունը ազդման կետից ամենայն հավանականությամբ ճառագայթվում նյութի միջով՝ նախընտրելով հետևելու ուղիներին, որոնք սահմանված են ինչպես երկարացափական մոտիկությամբ, այնպես էլ մեխանիկական համահումչությամբ սկզբնական ազդման հետ:

Սիմուլացիայի վերջնական արդյունքը ցանցի պատկերն է, որը համապատասխանում է կոտրված ուսանյակի նախշին (նկար 8-ի վերջնական պատկեր):

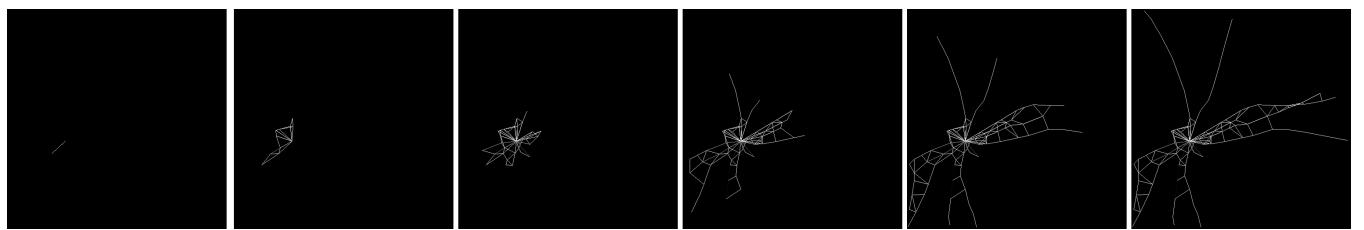


Figure 8: An animation of fracturing of a lens simulated by setting the stress field and applying PBR.

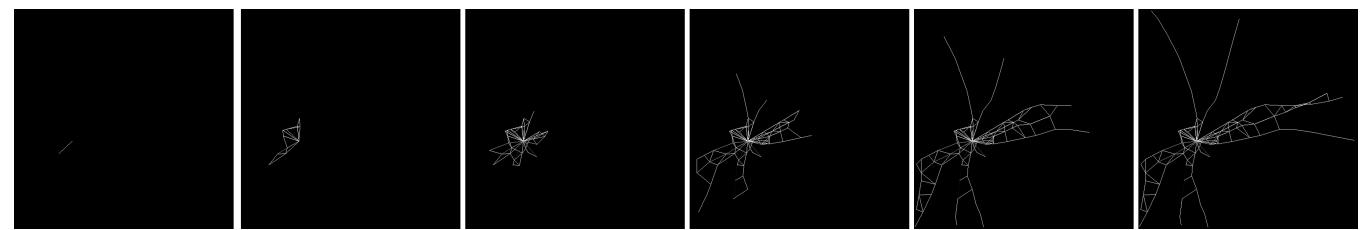


Figure 8: Ոսպնյակի ճեղքման անհմացիա, որը մոդելավորվել է սթրեսի դաշտի սահմանմամբ և PBR-ի կիրառմամբ R.

Static Experiment

In order to understand the effect of these fractures on the resultant images, we first conduct a indoor static experiment as referenced in Section Introduction. We use various tempered glass sheets for this experiment, which we break randomly using a small hammer with one single or multiple break points. Then, we place a 36 MP JVC GC-PX10 hybrid camera mounted on a tripod with a clamp in front for the tempered glass.

Fig. 9(a) shows the detailed setup with the camera mount and tempered glass held in place with a clamp. Fig. 9(b) shows the image captured by the camera and the Fig. 9(c) shows the single vehicle placed as the primary object being captured by the camera through the tempered glass. The scene is illuminated using overhead fluorescent lights.

Fig. 10 shows some of the fractures/scratched patterns on the tempered glass. These patterns were intentionally randomized, employing multiple focal points and different levels of force to mimic the unpredictable and varied nature of real-world glass damage. By applying diverse force strengths, we were able to produce a spectrum of fractures and scratches, ranging from fine surface abrasions to more pronounced fractures. This approach was chosen to closely replicate the types of damage that glass surfaces may encounter in actual conditions—such as those caused by impacts, debris, or environmental stressors—thereby ensuring the relevance and realism of our experimental setup. These representative damage patterns allow us to more effectively analyze the influence of glass imperfections on sensor performance and object detection algorithms.

Two different fracture patterns and their resultant images are shown in Fig. 11 and Fig. 12. We would like to note that we varied the focal lengths of the camera considerably to understand how the images look under near- and far-focus. The outputs show that even minor scratched patterns show up in the image output whereas much stronger multi-fracture pattern can blur almost the entire image. This experiment provides the intuition on which our simulation and visualization framework is built.

Increased AP for pedestrians in KITTI

We would like to point out that the increased AP for the pedestrian class was something that even we were surprised at first. However, a careful-qualitative deep-dive analysis helped us understand that this was occurring as a result of the glass cracks making it easier for the model to classify pedestrians because of enhanced edges around them. This wasn't an edge artifact but instead the glass crack acting as an additional edge boundary clearly separating the pedestrian and the background. A similar result was also observed in [1] where the overall AP was increased in adversarial images.



Figure 9: Experimental setup for collecting images impacted by scratched/broken outer layers for a camera. (a) shows the entire setup for taking adversarial images. (b) shows the position of the camera w.r.t. the scene being captured. (c) shows the scene being captured by the camera

Ստատիկ Փորձարկում

Այս ճեղքերի ազդցությունը ստացված պատկերների վրա հասկանալու համար մենք նախ կառարում ենք ներսի ստատիկ փորձարկում, ինչպես նշված է Բաժին Ներածություն-ում: Այս փորձարկման համար մենք օգտագործում ենք տարբեր կարծրացված ապակու թերթեր, որոնք պատահականորեն կո տրում ենք փորձ մուրճով՝ մեկ կամ մի քանի կոտորման կետերով: Այնուհետև, մենք տեղադրում ենք 36 ՄՊ JVC GC-PX10 հիբրիդային տեսախցիկը, որը ամրացված է եռոտանի վրա սեղմակով՝ կարծրացված ապակու դիմաց:

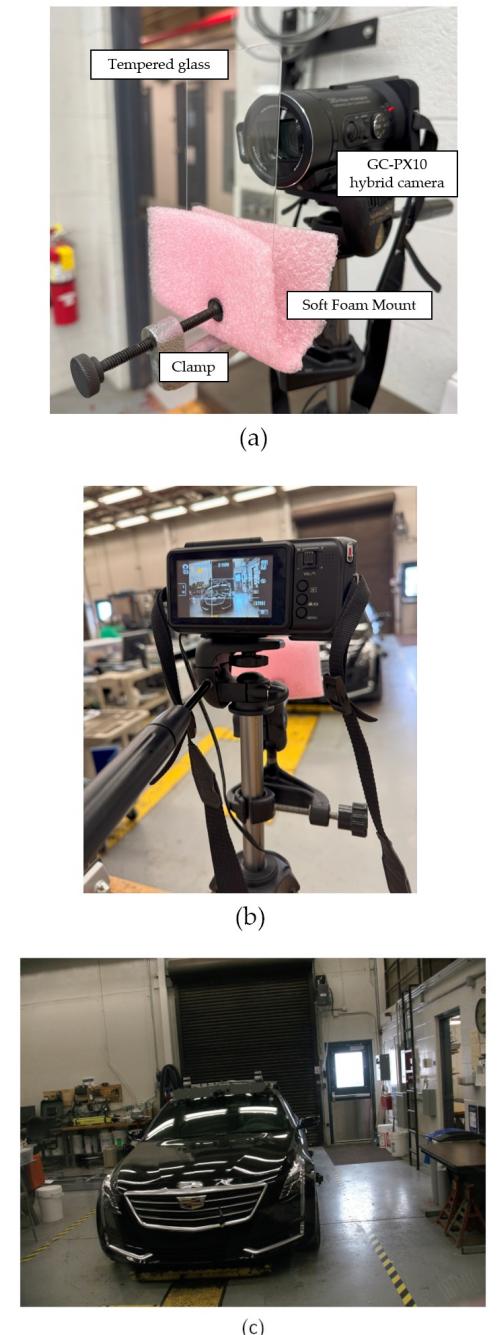
Նկ. 9(a)-ում ցուցադրված է մանրամասն տեղադրում՝ տեսախցիկի ամրակով և կարծրացված ապակին սեղմակով պահպան: Նկ. 9(b)-ում ցուցադրված է տեսախցիկի կողմից նկարահանած պատկերը, իսկ Նկ. 9(c)-ում ցուցադրված է մեկ մերենա, որը տեղադրված է որպես հիմն ական օբյեկտ, որը նկարահանվում է տեսախցիկի կողմից կարծրացված ապակու միջոցով: Տեսարանը լուսավորվում է վերևի ֆլուորեսցենտային լուսերով:

Նկ. 10-ում ցուցադրված է կարծրացված ապակու վրա որոշ ճեղքեր/բերձվածքների նախշեր: Այս նախշերը դիտավորյա պատահականացվել են՝ օգտագործելով բազմաթիվ ֆոկուսային կետեր և տարբեր ուժերի մակարակներ՝ նմանակելու հրական աշխարհի ապակու վնասավանքների անկանխատեսելի և բազմազան բնույթը: Տարբեր ուժերի կրառամամբ մենք կարողացան ստեղծել ճեղքերի և բերձվածքների սպեկտրը՝ սև ած նույր մակերեսային բերձվածքներից մինչև ավելի արտահայտված ճեղքեր: Այս մոտեցումը ընտրվել է՝ իրական պայմաններում ապակու մակերեսների վնասվածքների տեսակները մոտավորապես վերատրադրելու համար, ինչպիսիք են հաղողածների, առիջ կամ շրջակա միջավայրի սրբառությունի պատճենած վնասները, այդպիսով ապահովելով մեր փորձարկման տեղադրման համապատասխանությունն ու իրատեսությունը: Այս ներկայացուցական վնասավանքների նախշերը թույլ են տալիս մեզ ավելի արդյունավետ վերլուծել ապակու անկատարությունների ազդեցությունը սենսորների աշխատանքի և օրենսդրության հայտնաբերման այգրիթմների վրա:

Երկու տարբեր ճեղքաձիր նախշեր և դրանց արդյունքներն արտացոլված են Նկ. 11-ում և Նկ. 12-ում: Մենք ցանկանում ենք նշել, որ մենք զգայիրեն փոփոխել ենք տեսախցիկի ֆոկուսային հեռավորությունները՝ հասկանալու համար, թե ինչպես են պատկերները երևում մնուն և հեռու փոկուսում: Արդյունքները ցույց են տալիս, որ նույնին փորձածքների նախշերը հայտնվում են պատկերի արյունորում, մինչդեռ ավելի ուժեղ բազմաթիվ ճեղքածքների նախշը կարող է գրեթե ամբողջությամբ աղտօնանել պատկերը: Այս փորձը տրամադրում է այն հնատիղիքան, որի վրա հիմնված է մեր սիմուլացիայի վիզուալացիայի շրջանակը:

Հետիոտների համար AP-ի բարձրացում KITTI-ում

Մենք ցանկանում ենք նշել, որ հետիոտների դասի համար AP-ի բարձրացումը մեզ համար նույնպես անակնկալ էր սկզբում: Սակայն, մանրակրկիտ որակական վերլուծությունը օգնեց մեզ հասկանալու, որ սա տեղի էր ունենում ապակու ճեղքերի արյունորում, որոնք հեշտացնում էին մողեխին դասակարգել հետիոտներին՝ նրանց շուրջը նշեղացված եղբերի պատճառով: Սա եղի արտեֆակտ չէ, այլ ապակու ճեղքը հանդիս էր զային որպես լրացուցիչ եղանակ սահման, որը հստակ բաժանում էր հետիոտնին և ֆոնին: Նման արդյունքը նկատվեց նաև [1]-ում, որտեղ ընդհանուր AP-ն բարձրացավ հակառակորդական պատկերներում:



Նկար9: Փորձարական կարգադրում՝ բերձված/կոտորված արտաքին շերտերով ազդեցված պատկերներ հավաքելու համար: (ա) ցույց է տալիս ամրակով կարծրացված համարակարգային պատկերներ ստանալու համար: (բ) ցույց է տալիս տեսախցիկի դիմաց՝ նկարահանվող տեսարանի նկատմամբ: (ց) ցույց է տալիս տեսախցիկի կողմից նկարահանվող տեսարանը



Figure 10: Some fractures/scratched patterns on the glass we used for collecting the images. (a) A sharp force applied perpendicular to the glass surface, producing fractures occurring radially. (b) and (c) replicate a glass with scratches



Նկար 10: Որոշ ճաքեր/թերձվածքների նախշեր ապակու վրա, որը մենք օգտագործել ենք պատկերների հավաքագրման համար: (ա) Սուր ուժ կիրառված սողուակայաց ապակու մակերեսին, առաջացնելով ճաքեր, որոնք տարածվում են ճառագայթային: (բ) և (գ) կրկնօրինակում են թերձվածքներով ապակին

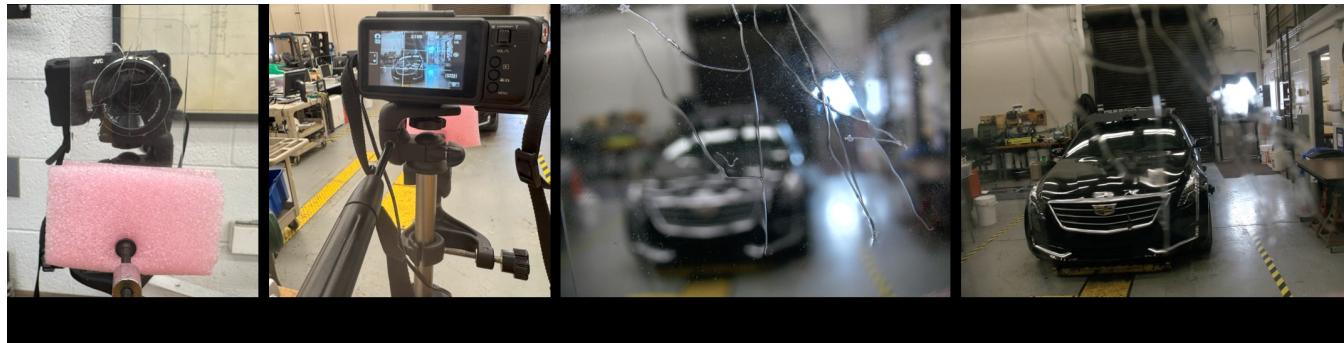
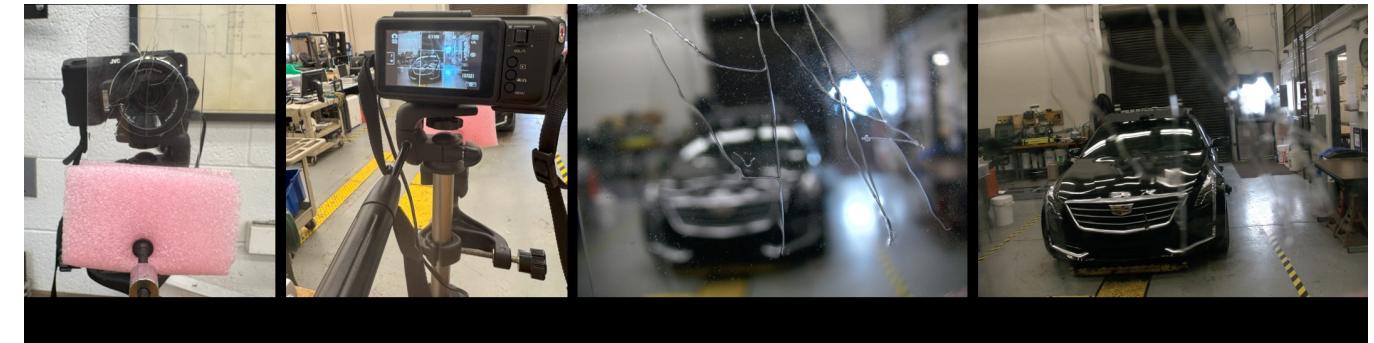


Figure 11: (a) Shows the scratched pattern placed in front of the camera, (b) shows the camera POV. (c) shows the image captured by the camera (short-focus). (d) shows the image captured by the camera (far-focus)



Նկար 11: (ա) Ցույց է տալիս թերձվածքների նախշը, որը տեղադրված է տեսախցիկի առջև, (բ) ցույց է տալիս տեսախցիկի տեսանկյունը: (գ) ցույց է տալիս տեսախցիկի կողմից նկարահանված պատկերը (կարճ ֆոկուս): (դ) ցույց է տալիս տեսախցիկի կողմից նկարահանված պատկերը (հեռու ֆոկուս)

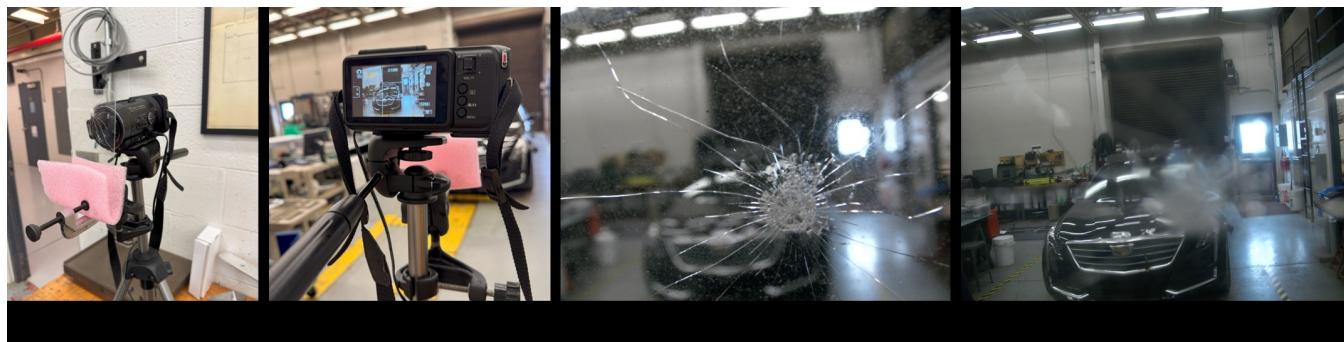


Figure 12: (a) Shows the broken glass pattern placed in front of the camera, (b) shows the camera POV. (c) shows the image captured by the camera (short-focus). (d) shows the image captured by the camera (far-focus)



Նկար 12: (ա) Ցույց է տալիս կոտրված ապակու նախշը տեսախցիկի առջև, (բ) ցույց է տալիս տեսախցիկի տեսանկյունը: (գ) ցույց է տալիս տեսախցիկի կողմից նկարահանված պատկերը (կարճ ֆոկուս): (դ) ցույց է տալիս տեսախցիկի կողմից նկարահանված պատկերը (հեռու ֆոկուս)

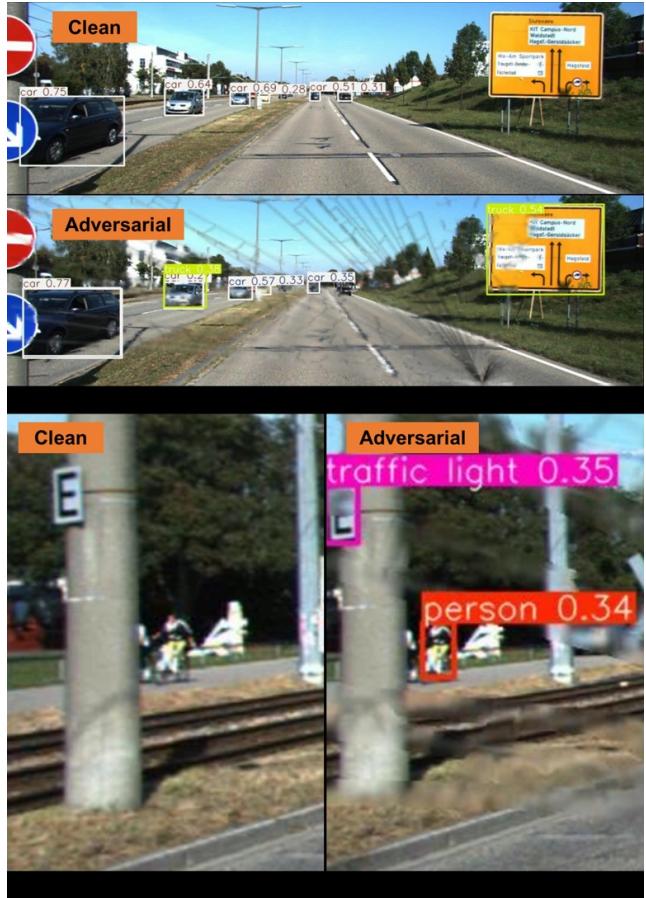
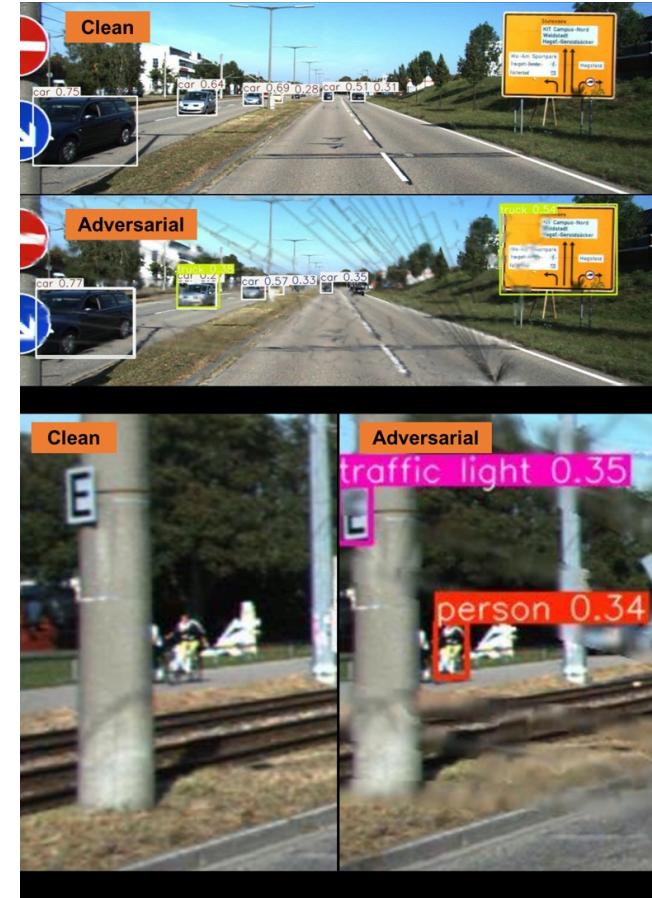


Figure 13: (a) Top - detections on a clean image; bottom - detections on an adversarial image.(b) YoLo fails to detect the person (c) Glass cracks allows the model to detect the person.



Նկար 13: (ա) Վերևում - հայտնաբերումներ մաքուր պատկերում; ներքևում - հայտնաբերումներ հակառակորդական պատկերում: (բ) YoLo-ն չի կարողանում հայտնաբերել անձին (գ) Ապակու ճաքերը թույլ են տալիս մոդելին հայտնաբերել անձին:

Dynamic Experiment

In this section, we describe the dynamic experiment mentioned in Section Introduction. We perform this experiment to understand the temporal perturbation introduced by a crack. We use a windshield crack of a vehicle and place a small camera on the dashboard behind the crack. Then we photograph two dynamic objects - a vehicle and a pedestrian as they move across the scene. Fig. 14 provides some specific image frames with inference from YOLOv8 for the vehicle class. We show that with the crack, the vehicle remains undetected in most frames. Additionally, almost every frame contains a false positive. Correspondingly, we present Fig. 15 as the frames with a person walking in the scene. We show that it intermittently provides detection and occasionally with a wrong class (surfboard).

Դինամիկ փորձարկում

Այս բաժնում մենք նկարագրում ենք դինամիկ փորձարկումը, որը նշված է Բաժին Ներածություն-ում։ Մենք կատարում ենք այս փորձարկումը՝ հասկանալու համար Ենթի առաջացրած ժամանակային խանգարումը։ Մենք օգտագործում ենք տրանսպորտային միջոցի առվելի ապակու ճեղք և փորձ տեսախցիկ տեղադրում ենք վահանակի վաս' Ենթի հետևում։ Այնուհետև լուսանկարում ենք երկու դինամիկ օբյեկտներ՝ տրանսպորտային միջոց և հետիւն, երբ նրանք շարժվում են տեսարանի միջով։ Նկ. 14-ը ներկայացնում է որոշակի պատկերային կադրեր՝ YOLOv8-ի միջոցով տրանսպորտային միջոցի դասի համար արված եզրակցություններով։ Մենք ցույց ենք տալիս, որ ճեղքի առկարգությամբ տրանսպորտային միջոցը մնում է չհայտնաբերված կադրերի մեջ մասում։ Բացի այդ, գրեթե յուրաքանչյուր կադր պարունակում է միայն դրական։ Համապատասխանաբար, մենք ներկայացնում ենք նկ. 15-ը՝ որպես կադրեր, որտեղ մարդը քայլում է տեսարանում։ Մենք ցույց ենք տալիս, որ այն ժամանակ առ ժամանակ ապահովում է հայտնաբերում և երբեմն միայն դասով (սերֆբորդ)։



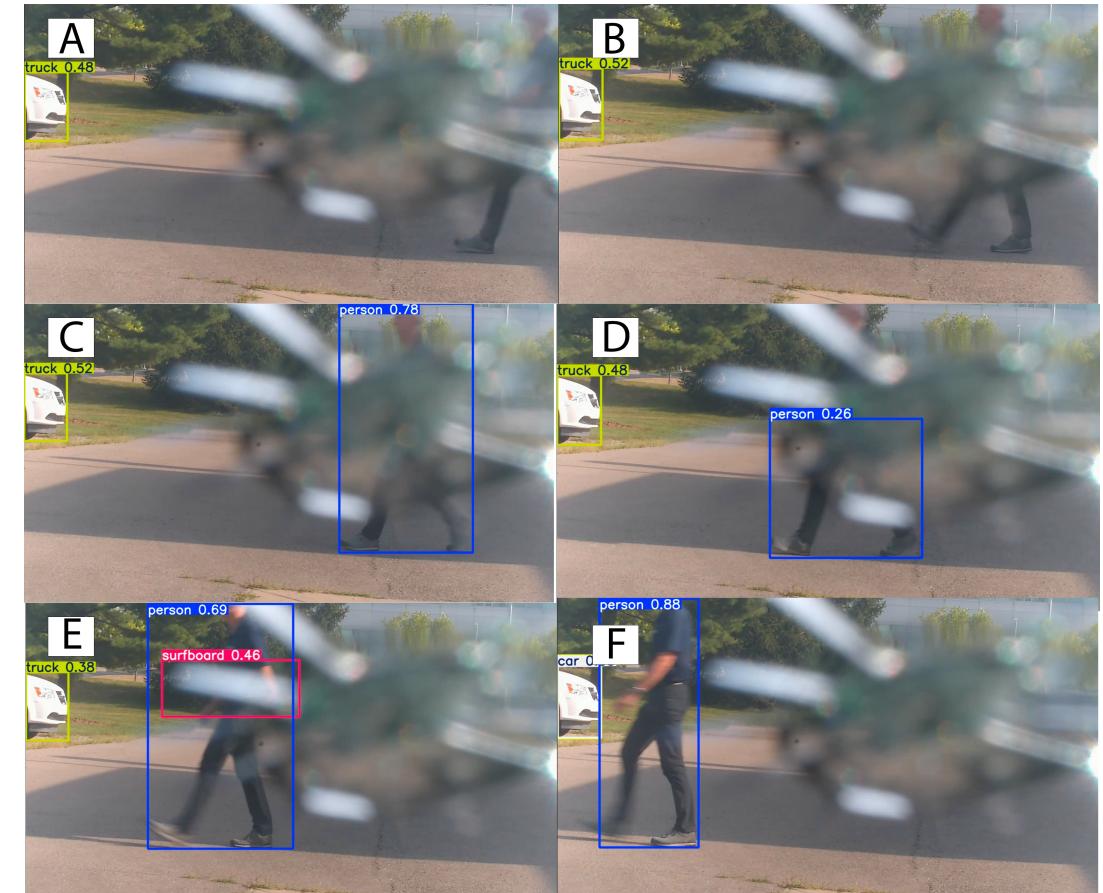
Figure 14: Specific frames of the images taken with the windshield crack with YOLOv8 inference for the vehicle class. A - false positive with no object in scene; B - no inference on vehicle; C - no inference on vehicle; D - first detection on vehicle; E - two different detections on the same vehicle; F - wrong bounding box area.



Նկար 14: Առցևի ապակու ճեղքով արված պատկերների որոշակի կադրեր՝ YOLOv8-ի միջոցով մեթնայի դասի համար: A - սխալ դրական՝ տեսարանում օբյեկտ չկա; B - մեթնայի վրա եզրակացություն չկա; C - մեթնայի վրա եզրակացություն չկա; D - մեթնայի առաջին հայտնաբերում; E - երկու տարրեր հայտնաբերումներ նույն մեթնայի վրա; F - սխալ սահմանափակող տուփի տարածք:



Figure 15: Specific frames of the images taken with the windshield crack with YOLOv8 inference for the person class. A - first entry of person in scene with no detection; B - no inference of person; C - first detection of person; D - partial detection of person; E - detection of person with other class; F - full detection of person.



Նկար 15: Առջևի ապակու ճեղքով արված պատկերների որոշակի կադրեր՝ YOLOv8-ի միջոցով անձի դասի համար: A - անձի առաջին մուտքը տեսարան՝ առանց հայտնաբերում չկա; B - անձի հայտնաբերում չկա; C - անձի առաջին հայտնաբերում; D - անձի մասնակի հայտնաբերում; E - անձի հայտնաբերում այլ դասի հետ; F - անձի ամբողջական հայտնաբերում:

Real glass fracture images

We present an example of the glass fracture images collected from the FreePik website overlaid on KITTI dataset along with YOLOv8 inference (Fig. 16). We show that the fracture removes some detections and decreases the detection confidence of others.

Իրական ապակու ճեղման պատկերներ

Մենք ներկայացնում ենք FreePik կայքից հավաքված ապակու կոտրվածքների պատկերների օրինակ, որոնք տեղադրված են KITTI տվյալների հավաքածովի վրա՝ YOLOv8-ի վերլուծությամբ (նկ. 16): Մենք ցուց ենք տախս, որ կոտրվածքը հեռացնում է որոշ հայտնաբերումներ և նվազեցնում է մյուսների հայտնաբերման վասահությունը:

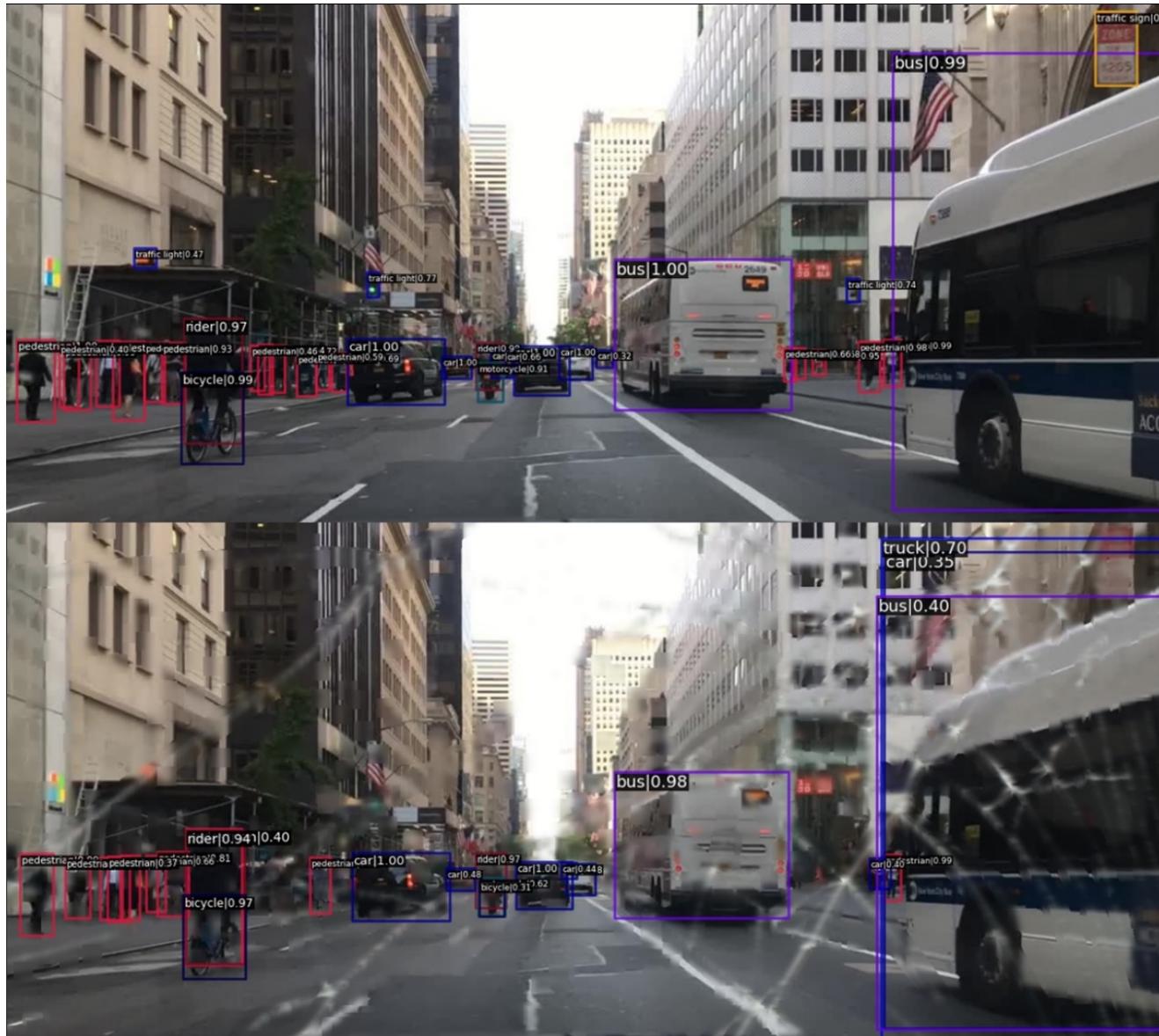
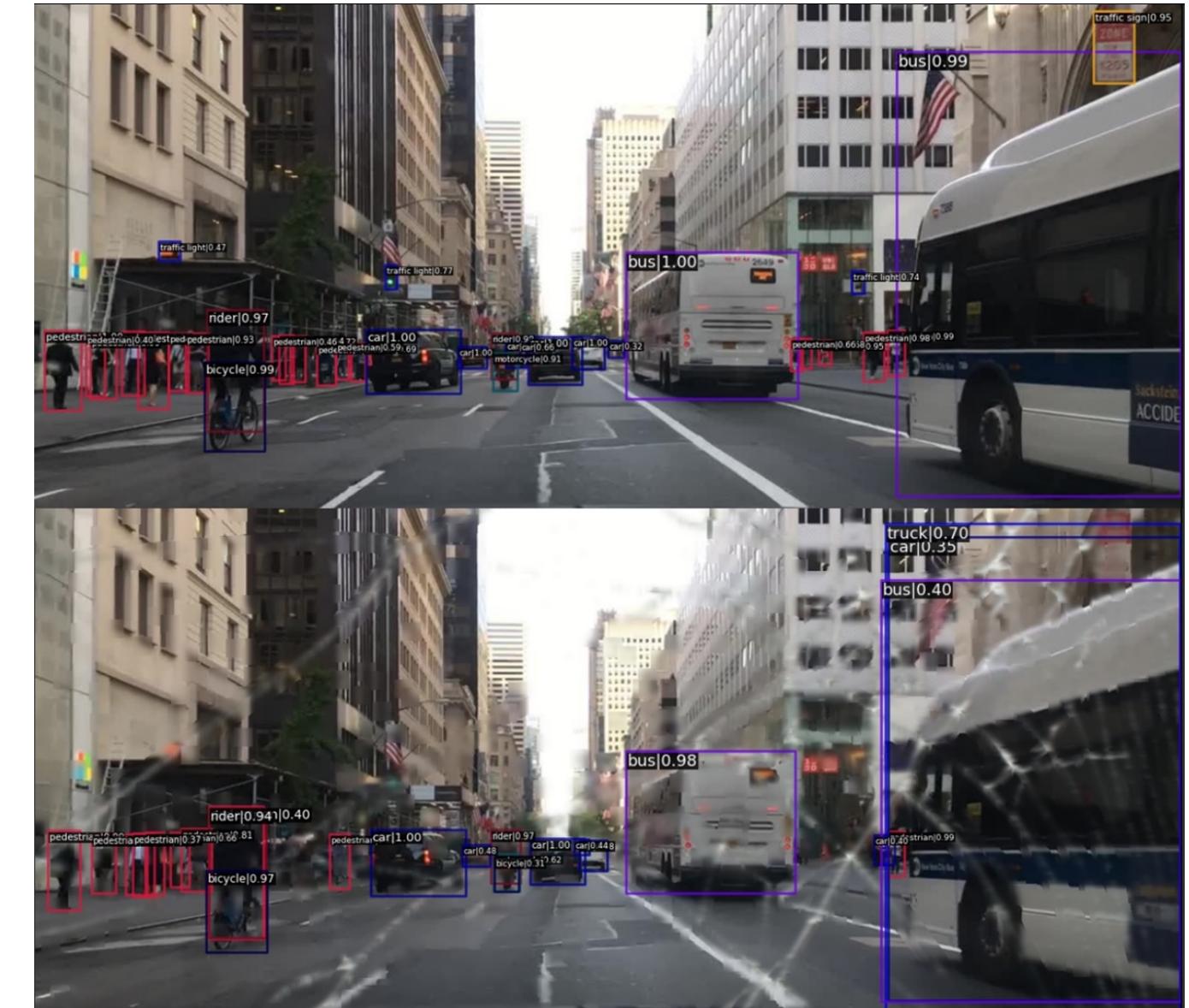


Figure 16: Top - Inference of PTv2 on a clean image from BDD100k. Bottom - Inference for a real broken glass image overlaid on BDD100k for comparison. We see two extra false positives in on the right side (truck, car) and several false negatives for the pedestrian class on the left of the adversarial image.



Նկար16: Վերևում՝ PTv2-ի եզրակացությունը մաքուր պատկերով BDD100K-ից: Ներքևում՝ իրական կոտրված ապակով պատկերով եզրակացությունը, որը դրված է BDD100K-ի վրա համեմատության համար: Մենք տեսնում ենք երկու լրացողի վիճակն առդրսներ աջ կողմում (բնօնատար, ավտոմեքենա) և մի քանի սխալ բացասական արդյունքներ հետհուտնի դասի համար՝ հակառակորդային պատկերի ձախ կողմում: