

# Verre fracturé, Caméras défaillantes : Simulation d'échantillons adversariaux basés sur la physique pour les systèmes de conduite autonome

**Manav Prabhakar, Jwalandhar Girnar, Arpan Kusari\*** Institut de recherche sur les transports de l'Université du Michigan 2901 Baxter Road, Salle 202, Ann Arbor, MI-48103 {prmanav, jwala, kusari}@umich.edu

## Résumé

Alors que de nombreuses recherches se sont récemment concentrées sur la génération d'échantillons adversariaux basés sur la physique, une catégorie critique mais souvent négligée provient des défaillances physiques au sein des caméras embarquées—composants essentiels aux systèmes de perception des véhicules automatisés. Les défaillances de la caméra, qu'elles soient dues à des contraintes externes provoquant une panne matérielle ou à des défauts de composants internes, peuvent directement compromettre la sécurité et la fiabilité des systèmes de conduite autonome. Premièrement, nous motivons l'étude en utilisant deux expériences réelles distinctes pour démontrer que les défaillances du verre entraîneraient effectivement l'échec des modèles de réseaux neuronaux basés sur la détection. Deuxièmement, nous développons une étude basée sur la simulation en utilisant le processus physique de la rupture du verre pour créer des scénarios perturbés, représentant une classe réaliste d'échantillons adversariaux basés sur la physique. En utilisant une approche basée sur le modèle par éléments finis (MEF), nous générerons des fissures de surface sur l'image de la caméra en appliquant un champ de contrainte défini par des particules dans un maillage triangulaire. Enfin, nous utilisons des techniques de rendu basé sur la physique (PBR) pour fournir des visualisations réalistes de ces fractures physiquement plausibles. Pour évaluer les implications en matière de sécurité, nous appliquons les effets de verre brisé simulés comme filtres d'image à deux jeux de données de conduite autonome—KITTI et BDD100K—ainsi qu'au jeu de données de détection d'image à grande échelle MS-COCO. Nous évaluons ensuite les taux d'échec de détection pour des classes d'objets critiques en utilisant des modèles de détection d'objets basés sur CNN (YOLOv8 et Faster R-CNN) et une architecture basée sur un transformateur avec des Pyramid Vision Transformers. Pour approfondir l'impact distributionnel de ces distorsions visuelles, nous calculons la divergence de Kullback-Leibler (K-L) entre trois distributions de données distinctes, en appliquant divers filtres de verre brisé à un jeu de données personnalisé (capturé à travers un pare-brise fissuré), ainsi qu'aux jeux de données KITTI et Kaggle de chats et chiens. L'analyse de la divergence K-L suggère que ces filtres de verre brisé n'introduisent pas de changements distributionnels significatifs. Notre objectif est de fournir une méthodologie robuste, basée sur la physique, pour générer des échantillons adversariaux qui reflètent les défaillances réelles des caméras, dans le but global d'améliorer la résilience et la sécurité des systèmes de conduite autonome contre de telles menaces physiques.

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

## Introduction

Les caméras sont omniprésentes en tant que capteurs à distance, collectant des données d'un environnement externe non strukturé et dynamique, souvent dans des conditions difficiles. Une défaillance ou un défaut dans un capteur est une divergence par rapport à l'état fonctionnel dans au moins un paramètre donné du système (van Schrick 1997). Ces défauts peuvent survenir en raison de causes internes (telles que l'usure) ou externes (température, humidité, etc.). Pour les caméras RGB, les causes internes incluent les pixels morts tandis que les causes externes incluent les boîtiers fracturés ou les lentilles extérieures, et la condensation. Ces défaillances soudaines sont difficiles à détecter et impactent négativement les algorithmes de détection d'objets—réduisant la précision et conduisant souvent à des hallucinations comme le montre la Fig. 1. Les défaillances survenant dans un véhicule automatisé (AV) par exemple, peuvent entraîner des problèmes de sécurité critiques entraînant des accidents et, dans certains cas, des décès.

Actuellement, à la connaissance des auteurs, il n'existe pas de méthodes rigoureuses pour générer des défaillances de capteurs basées sur des caméras (Ceccarelli et Secci 2022).

Dans ce travail, nous nous concentrons sur la défaillance du capteur due à des fractures dans tout verre recouvrant une caméra (ou un boîtier de caméra), bien que le processus détaillé dans cet article puisse être utilisé pour n'importe laquelle des défaillances de la caméra listées dans (Ceccarelli et Secci 2022). Ces effets de fracture du verre dans une caméra peuvent être causés par un objet externe frappant la caméra ou à la suite d'une montée soudaine de chaleur et/ou de pression à l'intérieur du boîtier. Dans le langage des réseaux neuronaux, une image capturée dans de telles conditions est considérée comme un échantillon adversarial. Des recherches antérieures (Akhtar et Mian 2018; Carlini et Wagner 2017; Szegedy et al. 2013) montrent que même de petites quantités de corruptions, parfois difficiles à percevoir par l'œil humain, suffisent à tromper complètement les réseaux neuronaux où un changement subtil des entrées peut entraîner un changement radical des sorties. Nous tenons à noter que (Li, Schmidt et Kolter 2019) ont fourni un paradigme d'attaque adversariale basé sur une caméra physique, qui constitue le travail le plus proche dans ce domaine. Ils ont présenté une modification de l'image en utilisant une superposition d'un autocollant translucide soigneusement conçu qui a conduit à une mauvaise classification.

Pour comprendre l'effet de ces fractures sur les images résultantes de la caméra, nous avons mené deux expériences distinctes : l'une

---

\*Auteur correspondant Copyright © 2026, Association pour l'avancement de l'intelligence artificielle ([www.aaai.org](http://www.aaai.org)). Tous droits réservés.

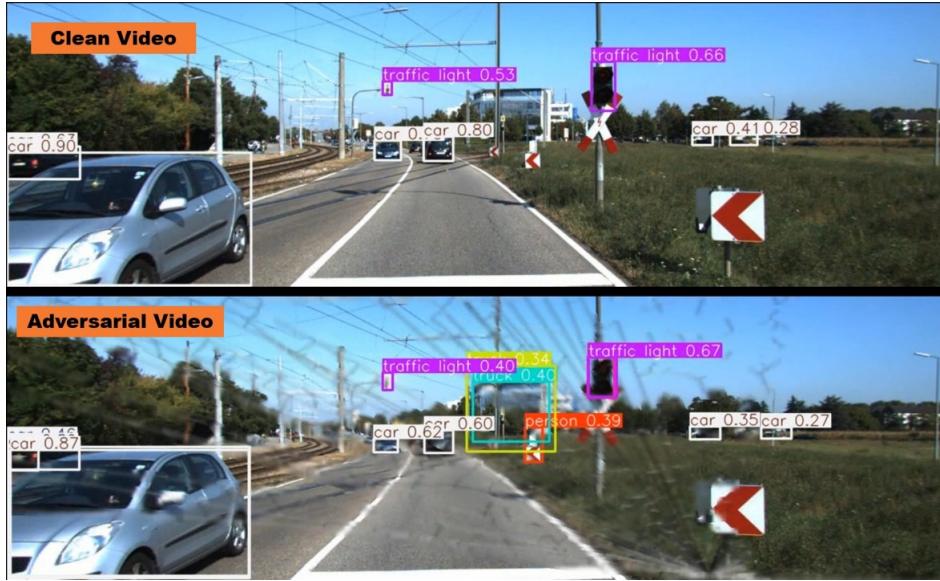


Figure 1 : Une comparaison qualitative entre une vidéo propre et une vidéo adversariale générée à l'aide de notre méthode de simulation et de rendu sur KITTI. Cette image montre des faux positifs et des niveaux de confiance réduits pour les vrais positifs. Référez-vous au matériel supplémentaire pour la vidéo complète.

dans un environnement statique intérieur et l'autre dans un environnement extérieur dynamique. La première consistait à fracturer du verre trempé et à le placer devant la caméra (voir Fig. 2(a)) avec un véhicule statique dans la scène pour comprendre comment différents motifs de fracture affectent la qualité et l'apparence de la scène. Nous avons capturé des images à différentes longueurs focales pour évaluer la variabilité de ces corruptions. Cela nous a aidés à répondre à certaines questions qualitatives sur l'apparence visuelle de ces fractures par rapport à leur étendue et leur intensité, motivant notre approche dans la section Plan focal et Simulation d'attaque physique. La configuration expérimentale et les résultats expérimentaux détaillés se trouvent dans la section Expérience statique du Supplémentaire. La deuxième expérience (Fig. 2(b)) consistait à enregistrer une vidéo en extérieur avec des véhicules dynamiques en conditions de lumière du jour en plaçant une caméra MobileEye à côté d'une fissure de pare-brise présentée dans la Fig. 2 (montrée en haut à gauche) et à effectuer une inférence en utilisant YOLOv8 (Jocher, Chaurasia, et Qiu 2023) pour obtenir une compréhension primitive de l'impact de tels scénarios sur les réseaux de détection d'objets. Nous avons observé que le modèle peut facilement détecter le véhicule dans une image propre tandis qu'il souffre d'échecs de détection (en bas à droite) ou génère de faux positifs (en bas à gauche). Fait intéressant, la présence d'une fissure peut également augmenter de manière inattendue la confiance dans la prédiction de la voiture présentant un bord clairement défini (0,92 en bas à gauche contre 0,75 en haut à gauche). Les résultats d'inférence détaillés avec la classe véhicule et personne sont donnés dans la section Expérience dynamique du Supplémentaire.

Nous avons ensuite cherché des images réelles de verre brisé en ligne (Sec. Images de fracture de verre réel du Supplémentaire) mais n'avons pas réussi à constituer un ensemble de données suffisamment grand pour permettre une approche basée sur les données pour la défense contre ces conditions adverses. De plus, nous avons expérimenté avec des outils CGI comme Maya et Blender pour

créer de tels effets, mais ils manquent de flexibilité, de contrôle, d'échelle et de physique pour simuler ces conditions. L'option de simulation la plus proche dans la littérature existante est ArcSim (Pfaff et al. 2014). Cependant, leurs sorties de simulation haute résolution sont extrêmement lentes ( $\approx 20$  heures), ce qui rend difficile la mise à l'échelle. En conséquence, nous avons orienté nos efforts vers la création d'un pipeline de simulation évolutif pour générer des fractures pouvant être utilisées pour faire progresser la pile de perception.

Pour une fracture de verre, le point principal, la force et l'angle d'incidence peuvent être aléatoires, mais la propagation et le motif résultant suivent un processus intrinsèquement physique (soit linéaire, soit radial). Nous construisons donc une simulation de fracture basée sur des particules dans un maillage triangulaire généré aléatoirement et effectuons la propagation du stress à travers le maillage. Notre simulation nous permet de produire les fractures au sein d'un maillage triangulaire à chaque état temporel discret  $\delta t$ . Nous utilisons OpenCV pour convertir le maillage donné en une image de motif de verre brisé correspondante. Nous utilisons ensuite le rendu basé sur la physique (PBR) (Pharr, Jakob, et Humphreys 2023) pour rendre de manière réaliste les fractures de surface en utilisant la fonction de distribution de la réflectance bidirectionnelle (BRDF) en calculant la quantité de lumière réfléchie à partir d'un point donné sur une surface en raison de la source ou des sources de lumière qui y sont incidentes.

En combinant notre approche de rendu avec trois ensembles de données open source populaires - KITTI (Geiger et al. 2013), BDD100k (Yu et al. 2020) et MS-COCO (Lin et al. 2014), nous sommes capables de générer des images adversariales de manière efficace. Un processus courant pour tester les images adversariales générées est de trouver le nombre de faux positifs/négatifs à travers l'espace d'image. Cependant, dans notre cas, en raison de l'effet adversarial étant local, nous ne pouvons pas nous fier simplement à une mesure basée sur l'image. Nous utilisons donc les images adversariales (similaires à la figure en bas à gauche de la Fig. 2) et extrayons les objets

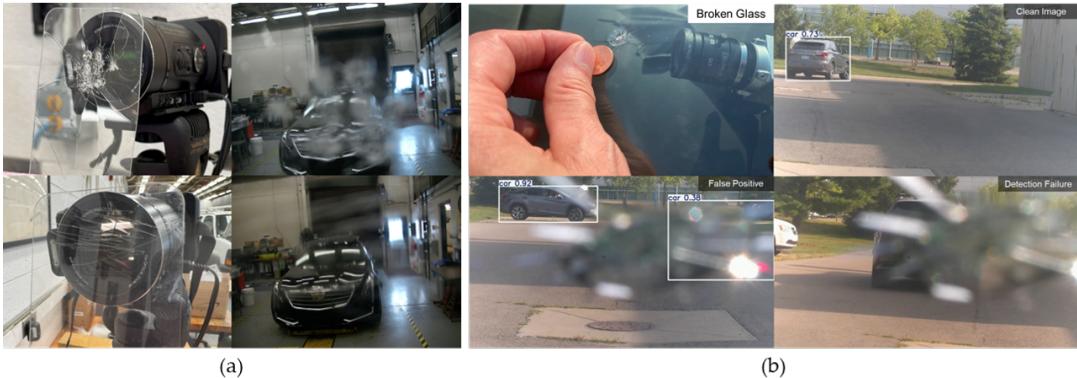


Figure 2 : (a) Expérience statique en intérieur. À gauche : Caméra avec 2 motifs différents de verre trempé fracturé ; à droite - images du véhicule sous les différentes fractures. (b) Expérience dynamique en extérieur. En haut à gauche - une fissure de la taille d'une pièce de monnaie sur le pare-brise ; en haut à droite - image nette avec le véhicule détecté à l'aide de YOLOv8 ; en bas à gauche - faux positif à travers la fissure ; en bas à droite - échec de détection à travers le verre. D'autres exemples de ces expériences sont fournis dans le matériel supplémentaire.

qui se trouvent dans la région où la fracture existe en utilisant les boîtes englobantes de vérité terrain. Nous utilisons ensuite YOLOv8, Faster R-CNN (Ren et al. 2016) et Pyramid Vision Transformer (PVTv2) (Wang et al. 2022) pour trouver le pourcentage d'objets qui échouent lorsque les filtres adversariaux sont appliqués. Nous fournissons également des études d'ablation pour comprendre les différences de distribution entre les trois ensembles d'images : images de verre brisé réelles collectées expérimentalement, images de verre brisé réelles collectées en ligne et les images générées. Nous calculons la divergence de Kullbeck-Liebler (K-L) pour ces distributions d'images afin de prouver la similarité des images générées avec les images de verre brisé réelles. Nous utilisons des images de chats du jeu de données Kaggle Cats and Dogs comme contrôle pour comprendre la différence entre les distributions d'images (PK).

Les principales contributions de l'article peuvent être résumées comme suit :

- Nous proposons une nouvelle manière d'abstraire la fracture du verre grâce à une combinaison de méthodes de propagation du stress et d'arbres couvrants minimaux, pour générer des motifs de verre brisé physiquement cohérents.
- Nous présentons une approche PBR pour faciliter un rendu réaliste des défaillances de la caméra qui peut être utilisé avec tout type de jeux de données de vision par ordinateur existants - à la fois images et vidéos.
- Nos pipelines de simulation et de rendu sont évolutifs et efficaces sur le plan computationnel ( $\approx 1.6s$ ), permettant leur utilisation par le milieu académique et l'industrie pour améliorer la robustesse et la protection hors distribution pour une large gamme d'applications.

## Contexte

### Échantillons adversariaux basés sur la physique

Le problème de l'échantillon adversarial peut être défini comme suit : pour un modèle  $M$  qui classe correctement un échantillon d'entrée  $X$  dans sa classe désignée, c'est-à-dire  $M(X) = y_{true}$ , l'ajout d'une erreur  $\epsilon$  à l'échantillon d'entrée  $X$  résulte en un échantillon modifié  $X'$  tel que  $M(X') \neq y_{true}$ . Ainsi, l'injection de l'erreur  $\epsilon$  résulte en un échantillon adversarial qui fait échouer le modèle.

Bien que l'idée de manipulation adversariale du modèle ait été identifiée dans le contexte de l'apprentissage automatique il y a déjà un certain temps (Dalvi et al. 2004), au cours de la dernière décennie, l'accent a été mis principalement sur les attaques adversariales contre les réseaux neuronaux (Szegedy et al. 2013; Goodfellow, Shlens, et Szegedy 2014). Dans ces articles, les chercheurs ont montré qu'une petite injection ciblée de bruit, presque imperceptible à l'œil humain, changeait complètement les étiquettes (Szegedy et al. 2013) et, inversement, des images pouvaient être générées qui semblaient totalement méconnaissables pour les humains mais qui avaient des classifications parfaites par les DNNs (Nguyen, Yosinski, et Clune 2015).

Bien que ces échantillons adversariaux testent le modèle pour d'éventuelles défaillances, ils manquent de réalisme physique dans leur génération et nécessitent un accès au modèle. Pour remédier à cela, certaines recherches récentes ont visé à construire des échantillons adversariaux physiquement pertinents. L'une des premières incursions dans ce domaine a été réalisée par (Kurakin, Goodfellow et Bengio 2018) qui ont ciblé la précision des modèles dans le monde physique en fournissant des images bruitées d'un appareil photo de téléphone portable, ce qui a conduit le modèle à classer incorrectement une grande partie des échantillons. Dans le même ordre d'idées, (Eykholt et al. 2018) ont démontré que de véritables panneaux de signalisation peuvent être perturbés avec de simples autocollants physiques placés stratégiquement pour tromper presque parfaitement les algorithmes DL de pointe, même avec des changements de point de vue. D'autres chercheurs ont placé des images adversariales (Kong et al. 2020), des patches translucides sur la caméra (Zolfi et al. 2021) ou des surfaces LiDAR artificielles (Tu et al. 2020) pour générer des échantillons qui trompent les détecteurs d'objets. Bien que ces recherches antérieures utilisent la physique pour générer les échantillons, elles ne proviennent pas de la modélisation d'un processus physique rigoureux et nous visons à combler cette lacune dans ce travail.

### théorie du verre fissuré/fracturé

Le sujet de la façon dont le verre se brise et se propage est encore une question de recherche ouverte et controversée, avec plusieurs théories physiques proposées (Rouxel et Brow 2012). Alors que la procédure microscopique de la fissuration du verre est débattue, au niveau macroscopique, la fissuration

dynamique est bien comprise. (Liu et al. 2021) ont analysé le processus de fissuration des lentilles en verre dans l'application de moulage de précision du verre en utilisant le MEF avec un modèle tridimensionnel dans un logiciel de simulation physique. Les paramètres physiques ont été entrés dans le logiciel et les chemins de fissure ont été analysés à l'aide des résultats de la simulation. Les auteurs ont effectué une simulation de température et de contrainte d'un modèle de maillage tridimensionnel de haute précision du verre moulé. (Iben et O'Brien 2009) ont proposé une méthode pour générer des fractures de surface dans une variété de matériaux, y compris le verre. Comme mentionné dans l'introduction, (Pfaff et al. 2014) ont fourni la simulation de la rupture du verre sous forme de feuille mince, ce qui constitue le travail le plus proche de notre méthode proposée.

### Méthodologie

Générer des défaillances réalistes du verre nécessite de créer des simulations à grande échelle basées sur la physique en résolvant la dynamique de fracture sur un maillage d'éléments finis triangulé avec les propriétés du verre.

### Simulation de verre brisé

Nous représentons le verre en utilisant des particules échantillonées à partir d'une distribution uniforme répartie sur un plan contraint sous la forme d'un maillage 2D en utilisant la triangulation de Delaunay contrainte. Cela élimine les triangles mal formés et évite les bords inégaux et irréalistes.

Chaque particule  $p_i$  a une position  $x_i$  et a des voisins les plus proches  $k_i$  dans un rayon  $r$  qui ont des arêtes existantes avec  $p_i$ . Mathématiquement, le maillage de triangulation  $\mathcal{M}$  représente un ensemble fini de 2-simplices tel que si

$$\forall (K, K') \in \mathcal{M} \times \mathcal{M}, |K| \cap |K'| = |K \cap K'|. \quad (1)$$

Les motifs de fissures dans le verre se produisent en raison du stress causé par la force externe ( $F$ ) au point d'impact initial  $p_I$  en s'upposant une loi de déformation spécifique (élasticité et plasticité) du verre ( $G$ ) (Kuna 2013). Nous calculons ensuite les paramètres de résistance sous forme de contrainte effective  $\sigma_V$  au point d'impact ( $V$ ) comme l'état de contrainte du point d'impact. Les valeurs de contrainte critique pour la résistance du verre  $\sigma_C$  sont trouvées en utilisant des tests sur des échantillons simples avec des conditions de chargement élémentaires (par exemple, test de traction). La fracture se produit alors lorsque la contrainte effective est supérieure à la contrainte critique divisée par le facteur de sécurité ( $S$ ) :

$$\sigma_V(G, F) > \frac{\sigma_C}{S}. \quad (2)$$

D'après la théorie classique de la résistance des matériaux, nous savons que la rupture dans la plupart des cas est contrôlée par les contraintes principales  $\sigma_I$  et  $\sigma_{II}$  pour les éléments 2D. La fissure initiale se produit soit par la fissure normale-plan qui se situe perpendiculairement à la direction de la contrainte principale la plus élevée  $\sigma_I$  (Rankine 1857), soit par la fissure de cisaillement-plan où les faces de fracture coïncident avec les plans d'intersection de la contrainte de cisaillement maximale  $\tau_{max} = (\sigma_I - \sigma_{II})/2$  (Coulumb 1776). Dans le cas du verre, nous supposons que la fracture initiale se produit perpendiculairement à la direction de la contrainte principale maximale.

Depuis le point d'impact initial  $p_I$ , la propagation du stress à travers le verre est instable car la fissure se développe brusquement sans qu'il soit nécessaire d'augmenter la charge externe. Depuis  $p_I$ ,

le stress se propage dans le voisinage du sommet  $k_i$  alors que le stress le long de la direction  $\overrightarrow{p_I p_j}$  où  $p_j \in k_i$  alors

$$\sigma_{p_j} = \sigma_V * \frac{\overrightarrow{p_I p_j} \cdot \vec{n}}{|\overrightarrow{p_I p_j}| |\vec{n}|}. \quad (3)$$

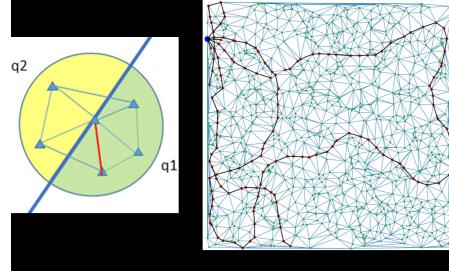


Figure 3 : (a) Pour un plan de séparation donné en bleu, la contrainte positive totale  $q_1$  et la contrainte négative totale  $q_2$  sont comparées et la propagation se produit du côté avec la contrainte totale la plus élevée et le noeud choisi qui est le plus proche du plan de séparation (indiqué en rouge). (b) Montre comment nous simulons une fracture dans un maillage à partir de son point d'impact (marqué en bleu) vers les noeuds subissant une contrainte au-delà de leur résistance seuil (marqué en rouge).

Avec le stress calculé pour chaque arête, le stress positif total (montré dans la Fig. 3) peut alors être donné comme suit :

$$q_1 = \int_{\partial\Omega} \sigma_{p_j} \mathbb{I}(\sigma_{p_j} > 0) dA \quad (4)$$

pour la surface continue  $\Omega$  où se trouve la fonction indicatrice. Le stress positif total pour les simples discrets dans la zone correspondante A de rayon R est donné comme

$$q_1 = \sum_{K \in A_R} \sigma_K \mathbb{I}(\sigma_K > 0). \quad (5)$$

De même, le stress négatif total  $q_2$  est calculé. Ensuite, pour une magnitude plus grande  $\max(|q_1|, |q_2|)$ , nous choisissons l'arête correspondante avec la plus haute concentration de stress dans le segment donné comme le plan de séparation optimal, car cela fournit le maximum de soulagement de stress. Ainsi, le stress se propage le long des arêtes du maillage, dissipant le stress à chaque noeud.

L'application récursive de la propagation du stress est exécutée jusqu'à la convergence du stress dans tous les états, c'est-à-dire  $\sigma_p^{(t)} \simeq \sigma_p^{(t-1)} \forall p \in V$ .

La propagation du stress dans toutes les directions à travers tous les noeuds entraîne des fissures en retour comme expliqué dans (O'Brien et Hodgins 1999). Pour l'éviter, nous ne propagons que le long des arêtes où les niveaux de stress sont maximaux, mais effectuons une mise à jour du stress sur tous les noeuds voisins. Nous utilisons ensuite un arbre couvrant minimal (ACM) sur un maillage créé à l'aide de ces noeuds stressés. Nous combinons cet ACM avec notre champ de propagation de stress initial le long des arêtes pour calculer le motif final de fissure. L'ACM est une abstraction efficace car il connecte les noeuds qui sont plus proches les uns des autres et dans le champ de stress élevé tout en éliminant les redondances.

Notre processus de calcul de la propagation du stress est défini dans l'Algorithm 1 dans le Supplémentaire.

## Rendu basé sur la physique

Une fois que nous avons générée les fractures au niveau du maillage, notre objectif suivant est de créer un rendu visuel de ces fractures. Comme toutes les techniques de PBR, notre méthode est basée sur la théorie des micro-facettes qui stipule que toute surface peut être décrite par de minuscules miroirs parfaitement réfléchissants appelés micro-facettes (Pharr, Jakob, et Humphreys 2023).

Conformément à la théorie des micro-facettes et à la conservation de l'énergie, nous utilisons l'équation de réflectance,

$$L_o(x, \omega_o, \lambda, t) = L_e(x, \omega_o, \lambda, t) + L_r(x, \omega_o, \lambda, t) \quad (6)$$

où  $L_o(x, \omega_o, \lambda, t)$  est la radiance spectrale totale de longueur d'onde  $\lambda$  dirigée vers l'extérieur le long de la direction  $\omega_o$  à l'instant  $t$ , depuis une position particulière  $x$ .  $\omega_o$  est la direction de la lumière sortante.  $t$  est le temps.  $L_e$  est la radiance spectrale émise et  $L_r$  est la radiance spectrale réfléchie.

Soit  $I_1$  la fonction de distribution bidirectionnelle de réflectance,

$$I_1 = f_r(x, \omega_i, \omega_o, \lambda, t)$$

et soit  $I_2$  la radiance spectrale entrant vers  $x$  depuis la direction  $\omega_i$  à l'instant  $t$ .

$$I_2 = L_i(x, \omega_i, \lambda, t)$$

Alors,  $L_r$  peut être défini comme

$$L_r(x, \omega_o, \lambda, t) = \int_{\Omega} I_1 \cdot I_2 \cdot (\omega_i \cdot \mathbf{n}) d\omega_i \quad (7)$$

où  $\Omega$  est l'hémisphère unitaire centré autour de la normale de surface  $\mathbf{n}$  sur  $\omega_i$  tel que  $\omega_i \cdot n > 0$ .

En abstrayant l'équation de réflectance, nous visons à créer un rendu visuel de notre maillage de verre brisé. Nous avons  $L_e = 0$  car le verre n'émet pas de lumière. Maintenant, pour calculer  $L_r$ , nous considérons toute fissure entre les nœuds comme une micro-facette. Ensuite, nous pouvons définir  $L_r$  pour chaque fissure comme :

$$L_r = L_i(\omega_i \cdot \hat{\mathbf{n}}) \quad (8)$$

Étant donné les vecteurs unitaires ( $\hat{\omega}_o$ ) et ( $\hat{\omega}_i$ ) correspondant respectivement aux angles d'azimut ( $\alpha$ ) et de zénith ( $\theta$ ), nous calculons l'énergie moyenne incidente sur la fissure comme

$$\mathbb{E}(L_r) = \frac{|\hat{\omega}_\alpha \cdot \hat{n}_i| + |\hat{\omega}_\theta \cdot \hat{n}_i|}{2} \quad (9)$$

où  $\hat{n}_i$  est la normale de surface unitaire de la fissure.

Soit  $(I_r, I_g, I_b)$  l'intensité moyenne de la source lumineuse. Alors l'intensité de la fissure,  $I_c$  est définie comme

$$I_c = (I_r, I_g, I_b) \cdot \frac{\mathbb{E}(L_r)}{\sum L_r} \quad (10)$$

**Plan focal et simulation d'attaque physique** Bien que nous soyons capables de simuler des fractures réalistes, l'utilisation principale de notre travail est de pouvoir générer des exemples simulés superposés sur des ensembles de données existants (KITTI, BDD100k, MS-COCO) et de les comparer avec l'ensemble de données réelles sur route que nous avons créé.

Toute image capturée présentera des caractéristiques nettes des objets dans son plan focal. L'enceinte en verre recouvrant la caméra est extrêmement proche et n'est donc pas partie du plan focal.

Lorsque la fissure se produit, les rayons lumineux rebondissent de manière inégale le long de la fissure et créent un flou (exemple fourni dans la Fig. 4). Nous créons un masque binaire basé sur le motif de la fissure, puis floutons les fractures superposées sur l'image. Cela produit une image à mise au point éloignée. Pour une image à mise au point rapprochée, nous floutons l'image et nous concentrons sur le premier plan, c'est-à-dire la fissure.

## Expérimentation

### Jeu de données

Nous évaluons deux types de motifs de verre brisé - réel et simulé - sur trois ensembles de données open-source populaires - KITTI (Geiger et al. 2013), BDD100k (Yu et al. 2020) et MS-COCO (Lin et al. 2014). Les deux premiers représentent un domaine spécifique de conduite autonome tandis que le dernier est un ensemble de données d'images à usage général. Les images de motifs de verre brisé réels sont collectées sur le site FreePik<sup>1</sup> et représentent la référence dans notre cas. Nous avons collecté un total de 65 images et les avons étendues à un ensemble de 10,000 images via l'augmentation d'images en utilisant des décalages aléatoires, des retournements d'images et des techniques de recadrage. Nous générions également 10,000 images en utilisant notre simulateur physique. Nous superposons ensuite ces motifs de verre fissuré en utilisant notre pipeline PBR sur chaque image de validation des ensembles de données et collectons les résultats agrégés. Nous utilisons trois architectures de modèles YOLOv8, Faster R-CNN et PVTv2 avec des poids pré-entraînés pour générer des résultats de détection d'objets.

### Mise en œuvre

Notre modèle de simulation est développé en échantillonnant aléatoirement  $10^4$  particules à partir d'une distribution spatiale uniforme dans le cadre donné sur un CPU. Un arbre KD du package Python SciPy (Virtanen et al. 2020) utilisant les paramètres par défaut est construit pour trouver les voisins les plus proches approximatifs de chaque particule. Une triangulation de Delaunay est ensuite exécutée sur les particules pour créer un maillage triangulaire contraint. Nous utilisons une force d'impact de 500 unités avec un point d'impact aléatoire et un vecteur d'impact aléatoire. La propagation du stress se produit jusqu'à ce qu'un seuil de 300 unités soit atteint. Le PBR est effectué sur CPU en implémentant les méthodes décrites dans la section précédente en utilisant OpenCV et Python.

## Résultats et Discussion

Un changement majeur par rapport à la plupart des travaux précédents sur les exemples adversariaux est que nos motifs adversariaux générés n'affectent pas universellement tous les pixels d'une image. Par conséquent, la comparaison doit être effectuée uniquement pour la région de l'image où le motif existe. À cette fin, nous créons un masque binaire de chaque motif et produisons les résultats des objets qui existent uniquement dans ce motif.

Le Tableau 1 montre les résultats de la précision moyenne (AP) sous les images adversariales générées en utilisant les deux types de motifs de fissures (collectés en ligne et simulés) pour différentes classes. Pour KITTI, l'AP des autres classes diminue comme prévu avec la diminution de l'AP correspondant au pourcentage d'image occupé par la classe camion enregistrant la plus forte baisse. Pour BDD100K avec PVTv2-B0, nous constatons que la baisse de l'AP est la plus importante dans les images simulées mais globalement,

<sup>1</sup><https://www.freepik.com>



Figure 4 : (a) Montre l'image simulée avec la route et les véhicules dans le plan focal (PBR et mise au point lointaine). (b) désigne le motif de fissure simulé dans le plan focal (PBR et mise au point courte).

Tableau 1 : Précision moyenne (en pourcentage) des différentes classes dans KITTI, BDD100k et MS-COCO sous différentes images adversariales. x fournit la relation de superposition entre le jeu de données et le type de fissure de verre. Clean x Dataset - se réfère directement aux images particulières sans aucun échantillon adversarial. RO x Dataset - se réfère aux images réelles de verre fissuré collectées en ligne superposées sur des images propres. Sim x Dataset - se réfère aux motifs de fissures simulés superposés sur des images propres.

Jeu de données	Seuil IoU	Catégorie	Jeux de données propre x	Jeux de données RO x	Sim x Ensemble de données
KITTI (YOLOv8)	0.5	Piéton	25.64	69.72	<b>17.84</b>
		Camion	12.39	<b>3.59</b>	3.76
		Car	58.99	<b>50.7</b>	57.73
	0.75	Piéton	6.83	33.88	<b>6.02</b>
		Camion	11.29	<b>2.67</b>	2.79
		Car	31.25	<b>23.85</b>	30.15
BDD100k (PVTv2)	0.5	Piéton	66.47	54.33	<b>25.95</b>
		Camion	61.97	<b>52.83</b>	<b>52.02</b>
		Car	80.37	70.14	<b>56.78</b>
	0.75	Piéton	27.06	22.72	<b>10.60</b>
		Camion	47.03	<b>38.23</b>	42.52
		Car	46.23	45.97	<b>42.99</b>
MS-COCO (Faster R-CNN)	0.5	Personne	0.035	0.024	<b>0.024</b>
		Véhicules	2.14	<b>1.45</b>	1.87
		Food	35.34	<b>28.07</b>	30.65
	0.75	Personne	0.032	<b>0.022</b>	0.023
		Véhicules	1.56	<b>1.05</b>	1.07
		Food	24.59	<b>18.85</b>	22.00

la tendance se maintient avec la classe des piétons montrant la baisse la plus marquée. Pour MS-COCO, nous avons agrégé l'AP pour les super-catégories : personne, véhicules et nourriture. Cela est dû au fait que de nombreux objets dans MS-COCO occupent une plus petite surface dans le cadre de l'image, rendant difficile l'obtention de résultats significatifs pour toutes les catégories. Un résultat très intriguant est que la classe des piétons connaît une augmentation multiple de l'AP sous les motifs de verre brisé réels. Bien que cette tendance puisse sembler contre-intuitive, elle résonne avec les résultats de la Fig. 2 où la confiance de la voiture augmente en raison d'un bord. Cela montre en fait que l'AP dépend fortement du motif de fissure, rendant extrêmement important de créer des méthodologies de défense pour atténuer ces attaques adversariales.

### Études d'ablation

Nos résultats indiquent que les images simulées obtiennent un effet adversarial similaire à celui des images réelles. Ainsi, une étude d'ablation importante pour nous est de comprendre à quel point les motifs de fissures simulés se rapprochent des motifs de verre fissuré réels et de ceux collectés en ligne. Nous formons 5 distributions

- Motifs de fissures collectés en ligne (Fig. 5 en haut à gauche)
- Motifs de fissures simulés (Fig. 5 en bas à gauche)
- Motifs de fissures simulés superposés sur KITTI (Fig. 5 en bas à droite)
- Motifs de fissures collectés en ligne superposés sur KITTI (Fig. 5 en haut à droite)

Nous calculons maintenant la divergence K-L entre toutes ces distributions pour déterminer à quel point elles se ressemblent (voir Fig. 6). Afin de fournir un contrôle, nous comparons KITTI à des images de chats du jeu de données Kaggle, ce qui donne une divergence K-L de 2,434. Sur cette échelle, les images PBR de verre brisé ont une différence de 0,36 par rapport aux motifs réels de verre brisé, tandis que les filtres de verre brisé superposés sur les images KITTI ont une divergence K-L si modeste.

La Fig. 7 montre une analyse du temps de calcul pour chacun de nos modules et pour différents nombres de particules. Nous effectuons cette analyse sur 100 exécutions, générant des points d'impact aléatoires, des angles d'impact et une structure de maillage avec un nombre fixe de particules. La différence de temps de calcul pour différentes exécutions peut être attribuée au point d'impact et à l'angle d'impact.

• Jeu de données réels sur route (illustré dans la Fig. 2)

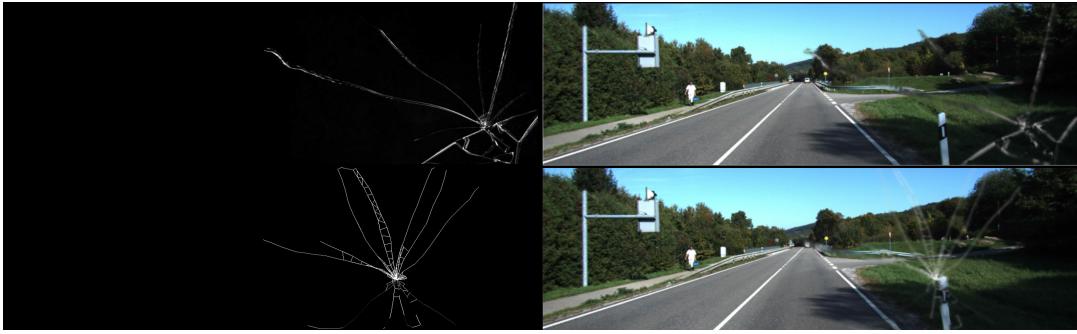


Figure 5 : En haut à gauche - Motif de fissure collecté en ligne sur Freepik ; en haut à droite - motif de fissure en ligne superposé sur KITTI ; en bas à gauche - motif de fissure simulé avec PBR ; en bas à droite - motif de fissure simulé superposé sur KITTI.

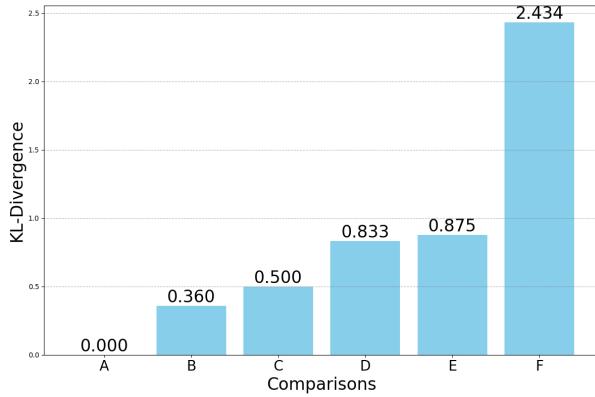


Figure 6 : Divergence K-L de différentes paires de distributions d'images. Jeux de données : RC - Jeu de données réel sur route (voir Fig. 2), KITTI et Chats. Filtres : RO - Réel (collecté en ligne) et Sim - Simulé. Divergence K-L entre (x - relation de superposition) : A - (Sim x KITTI) vs (Sim x KITTI) ; B - (Sim vs RO) ; C - (RC propre vs KITTI) ; D - (RC cassé) vs (RO x KITTI) ; E - (RC cassé) vs (Sim x KITTI) ; F - KITTI vs Chats.

L'angle d'impact. La visualisation des fissures et le temps de rendu varient également en raison des masques de tailles différentes formés par les motifs de fracture variables. Nous faisons également varier le nombre de particules et observons comment le temps d'exécution augmente avec l'augmentation du nombre de particules. Toutes ces exécutions ont été rendues sur des images du jeu de données KITTI avec des dimensions de  $(375 \times 1242 \times 3)$ .

### Conclusion et perspectives futures

Nous avons introduit une nouvelle classe d'échecs adversariaux résultant du processus physique de défaillances dans la caméra. Dans cet article, nous proposons une approche pour générer un motif de verre brisé réaliste à partir d'une simulation physique et l'intégrer ensuite dans des ensembles de données d'images existants en utilisant le rendu basé sur la physique. Nous montrons que les images adversariales simulées peuvent entraîner des erreurs significatives dans la détection d'objets.

Dans ce travail, nous abordons les attaques adversariales en boîte noire provenant de phénomènes physiques réels et naturels, non artificiellement conçus pour exploiter des vulnérabilités spécifiques du modèle.

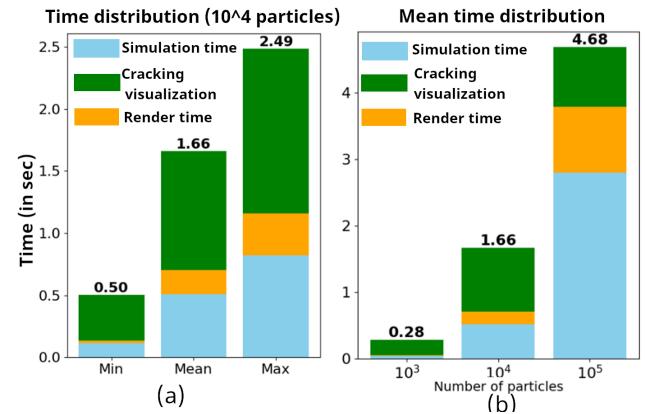


Figure 7 : (a) Temps moyen pris par différents modules de notre pipeline sur 100 exécutions. (b) Le temps minimum, maximum et moyen pris par différents modules sur 100 exécutions pour un maillage de particules  $10^4$ . Pour ces graphiques, nous présentons le temps pris pour la simulation (temps de simulation), la conversion du maillage en verre (visualisation de la fissuration) et enfin le rendu (temps de rendu).

Nous ne supposons aucune connaissance des attributs, des poids ou de l'architecture du modèle, garantissant que les attaques sont transférables entre divers modèles. Les méthodes adversariales physiques (Patch Translucide, RP2) peuvent toutes être qualifiées d'occlusions soit de la caméra, soit des objets capturés. L'adversarialité provient de l'effet de l'inférence du modèle dû à ces occlusions. Notre pipeline PBR mélange les fissures avec les images sources sous forme de motifs translucides et flous, impactant l'encodage de l'espace latent plutôt que de provoquer une occlusion directe, entraînant des détections incorrectes.

Bien que ce travail introduce une méthode basée sur la physique pour la génération de motifs de verre brisé spécifiquement, les défaillances de la caméra englobent d'autres effets tels que l'éblouissement solaire, la surexposition, la sous-exposition, la condensation, etc. Notre travail futur se concentrera sur la création d'une boîte à outils adversariale pour la génération réaliste de ces effets en utilisant la physique et, par la suite, les placer sur des ensembles de données d'images existants et des plateformes de simulation de voitures pour promouvoir davantage de recherches dans ce domaine des défaillances partielles de la caméra.

## Références

- Akhtar, N.; et Mian, A. 2018. Menace des attaques adversariales sur l'apprentissage profond en vision par ordinateur : Une enquête. *Ieee Access*, 6 : 14410–14430.
- Carlini, N.; et Wagner, D. 2017. Les exemples adversariaux ne sont pas facilement détectés : Contournement de dix méthodes de détection. Dans *les Actes du 10e atelier ACM sur l'intelligence artificielle et la sécurité*, 3–14.
- Ceccarelli, A.; et Secci, F. 2022. Défaillances des caméras RGB et leurs effets dans les applications de conduite autonome. *IEEETransactions on Dependable and SecureComputing*. Coulumb, C.-A. 1776. Essai sur une application des règles des maximis et minimis à quelques problèmes de statique relatifs, à l'architecture. *Mem. Acad. Roy. Div. Sav.*, 7: 343–387. Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; et Verma, D. 2004. Classification adversariale. Dans *les Actes de la dixième conférence internationale ACM SIGKDD sur la découverte de connaissances et l'exploration de données*, 99–108. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; et Song, D. 2018. Attaques physiques robustes sur la classification visuelle par apprentissage profond. Dans *les Actes de la conférence IEEE sur la vision par ordinateur et la reconnaissance de formes*, 1625–1634. Geiger, A.; Lenz, P.; Stiller, C.; et Urtasun, R. 2013. La vision rencontre la robotique : Le jeu de données KITTI. *International Journalof RoboticsResearch(IJRR)*. Goodfellow, I. J.; Shlens, J.; et Szegedy, C. 2014. Expliquer et exploiter les exemples adversariaux. *arXiv preprintarXiv:1412.6572*. Iben, H. N.; et O'Brien, J. F. 2009. Génération de motifs de fissures de surface. *GraphicalModels*, 71(6): 198–208. Jocher, G.; Chaurasia, A.; et Qiu, J. 2023. Ultralytics YOLO. Kong, Z.; Guo, J.; Li, A.; et Liu, C. 2020. Physgan : Génération d'exemples adversariaux résilients dans le monde physique pour la conduite autonome. Dans *les Actes de la conférence IEEE/CVF sur la vision par ordinateur et la reconnaissance de formes*, 14254–14263. Kuna, M. 2013. Éléments finis en mécanique de la rupture. *Solidmechanics andits applications*, 201: 153–192. Kurakin, A.; Goodfellow, I. J.; et Bengio, S. 2018. Exemples adversariaux dans le monde physique. Dans *la sécurité et la sûreté de l'intelligence artificielle*, 99–112. Chapman and Hall/CRC. Li, J.; Schmidt, F.; et Kolter, Z. 2019. Autocollants de caméra adversariaux : Une attaque physique basée sur la caméra contre les systèmes d'apprentissage profond. Dans *la conférence internationale sur l'apprentissage automatique*, 3896–3904. PMLR. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Doll'ar, P.; et Zitnick, C. L. 2014. Microsoft coco : Objets communs dans le contexte. Dans *Computer Vision–ECCV 2014 : 13e Conférence européenne, Zurich, Suisse, 6–12 septembre 2014, Actes, Partie V 13*, 740–755. Springer.
- Liu, Y.; Xing, Y.; Li, C.; Yang, C.; et Xue, C. 2021. Analyse de la fracture des lentilles dans le moulage de verre de précision avec la méthode des éléments finis. *Applied Optics*, 60(26) : 8022–8030.
- Nguyen, A.; Yosinski, J.; et Clune, J. 2015. Les réseaux neuronaux profonds sont facilement trompés : Prédictions de haute confiance pour des images non reconnaissables. Dans *les Actes de la conférence IEEE sur la vision par ordinateur et la reconnaissance de formes*, 427–436. O'Brien, J. F.; et Hodgins, J. K. 1999. Modélisation graphique et animation de la fracture fragile. Dans *les Actes de l'ACM SIGGRAPH 1999*, 137–146. ACM Press/Addison-Wesley Publishing Co. Pfaff, T.; Narain, R.; De Joya, J. M.; et O'Brien, J. F. 2014. Déchirure et fissuration adaptatives de feuilles minces. *Transactions ACM sur les graphiques (TOG)*, 33(4) : 1–9. Pharr, M.; Jakob, W.; et Humphreys, G. 2023. *Rendu basé sur la physique : De la théorie à l'implémentation*. MIT Press. PK, A. ??? Mini ensemble de données de chats et chiens Kaggle. h  
t  
t  
p  
s://  
w  
w  
w.  
[kaggle.com/datasets/aleemaparakatta/cats-and-dogs-mini-dataset](https://kaggle.com/datasets/aleemaparakatta/cats-and-dogs-mini-dataset). Consulté le : 2024-09-30. Rankine, W. J. M. 1857. II. Sur la stabilité des terres meubles. *Transactions philosophiques de la Royal Society de Londres*, (147) : 9–27. Ren, S.; He, K.; Girshick, R.; et Sun, J. 2016. Faster R-CNN : Vers la détection d'objets en temps réel avec des réseaux de propositions de régions. *Transactions IEEE sur l'analyse des formes et l'intelligence artificielle*, 39(6) : 1137–1149. Rouxel, T.; et Brow, R. K. 2012. L'écoulement et la fracture des verres avancés—un aperçu. *Journal international des sciences appliquées du verre*, 3(1) : 1–2. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; et Fergus, R. 2013. Propriétés intrigantes des réseaux neuronaux. *arXivpreprintarXiv:1312.6199*. Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; et Urtasun, R. 2020. Exemples adversariaux physiquement réalisables pour la détection d'objets par lidar. Dans *les Actes de la conférence IEEE/CVF sur la vision par ordinateur et la reconnaissance de formes*, 13716–13725. van Schrick, D. 1997. Remarques sur la terminologie dans le domaine de la supervision, de la détection de défauts et du diagnostic. *Volumes des Actes de l'IFAC*, 30(18) : 959–964. Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, I.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; et les Contributateurs de SciPy 1.0. 2020. SciPy 1.0 : Algorithmes fondamentaux pour le calcul scientifique en Python. *Nature Methods*, 17 : 261–272. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; et Shao, L. 2022. Pvt v2 : Améliorations des bases avec le transformateur de vision pyramidale. *Médias visuels computationnels*, 8(3) : 415–424.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; et Darrell, T. 2020. BDD100k : Un ensemble de données de conduite diversifié pour l'apprentissage multitâche hétérogène. Dans *les Actes de la conférence IEEE/CVF sur la vision par ordinateur et la reconnaissance de formes*, 2636–2645. Zolfi, A.; Kravchik, M.; Elovici, Y.; et Shabtai, A. 2021. Le patch translucide : Une attaque physique et universelle sur les détecteurs d'objets. Dans *les Actes de la conférence IEEE/CVF sur la vision par ordinateur et la reconnaissance de formes*, 15232–15241.

## Algorithme de propagation du stress

Algorithme 1 décrit la procédure pour simuler la propagation du stress à travers un matériau suite à un événement d'impact. L'algorithme prend en entrée la localisation de l'impact ( $pt$ ), l'amplitude de la force d'impact ( $F$ ), le vecteur de direction de l'impact ( $v$ ), et l'arête parente ( $PE$ ) associée au site d'impact. Il utilise également un rayon du plus proche voisin  $R$  pour déterminer l'ensemble des emplacements candidats pour la propagation du stress.

---

Algorithme 1 : Propagation du stress

---

```

1:  $pt \leftarrow$  Point d'impact
2:  $F \leftarrow$  Force d'impact
3:  $PE \leftarrow$  Arête parente
4:  $v \leftarrow$  Vecteur d'impact
5:  $R \leftarrow$  Rayon du plus proche voisin
6:
7: procédure PROPAGERSTRESS( $Pt, F, V, PE$ )
8:    $frontiers \leftarrow KDTTree - queryRadius(R)$ 
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:  PROPAGERSTRESS( $Pt, F, V, PE$ )
16: fin de la
procédure

```

---

Tout d'abord, il utilise une structure de données en arbre KD pour interroger efficacement tous les points (frontières) dans un rayon donné  $R$  du point d'impact. Pour chaque frontière, il calcule un vecteur directionnel unitaire du point d'impact vers la frontière ( $NN$ ). Il projette ensuite le vecteur d'impact  $v$  sur cette direction pour obtenir la similarité cosinus  $\cos(\theta)$ , capturant la relation angulaire entre la direction de l'impact et la direction de propagation candidate. Pour chaque candidat, la valeur résultante est utilisée, avec la force d'impact, pour calculer la contrainte correspondante à ce point. L'algorithme sélectionne ensuite le candidat avec la valeur de contrainte maximale. Le vecteur d'impact  $v$  et l'arête parente  $PE$  sont mis à jour pour correspondre à cette nouvelle direction. Le processus est répété de manière récursive, permettant à l'onde de contrainte simulée de se propager de manière itérative à travers le matériau le long du chemin de transfert de contrainte le plus élevé.

Cette approche vise à imiter la manière dont la contrainte d'un point d'impact est le plus susceptible de se propager à travers un matériau—suivant préférentiellement des chemins définis par la proximité géométrique et l'alignement mécanique avec l'impact initial.

Le résultat final de la simulation est la réalisation du maillage sous forme d'image qui correspond au motif de lentille brisée (image finale de la Fig. 8).

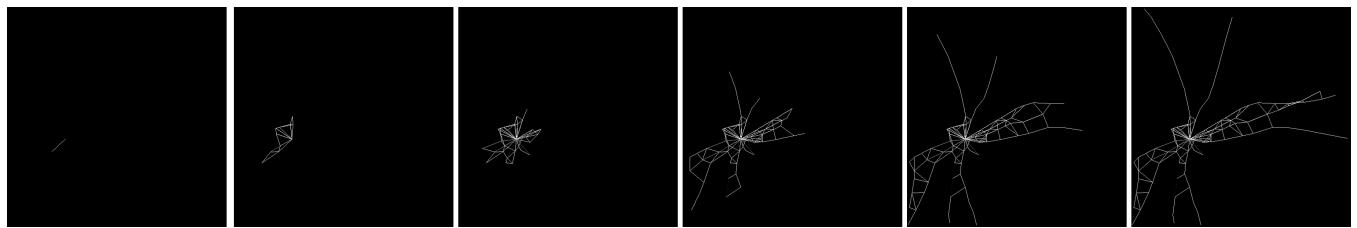


Figure 8 : Une animation de la fracture d'une lentille simulée en définissant le champ de contrainte et en appliquant PB R.

## Expérience Statique

Afin de comprendre l'effet de ces fractures sur les images résultantes, nous menons d'abord une expérience statique en intérieur comme référencé dans la Section Introduction. Nous utilisons diverses plaques de verre trempé pour cette expérience, que nous brisons aléatoirement à l'aide d'un petit marteau avec un ou plusieurs points de rupture. Ensuite, nous plaçons un appareil photo hybride 36 MP JVC GC-PX10 monté sur un trépied avec une pince devant le verre trempé.

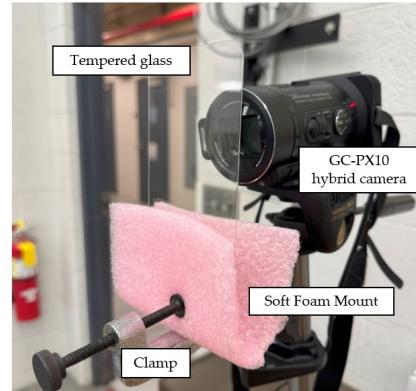
La Fig. 9(a) montre l'installation détaillée avec le support de la caméra et le verre trempé maintenu en place avec une pince. La Fig. 9(b) montre l'image capturée par la caméra et la Fig. 9(c) montre le véhicule unique placé comme l'objet principal capturé par la caméra à travers le verre trempé. La scène est éclairée par des lumières fluorescentes au plafond.

La Fig. 10 montre certains des motifs de fractures/rayures sur le verre trempé. Ces motifs ont été intentionnellement randomisés, en utilisant plusieurs points focaux et différents niveaux de force pour imiter la nature imprévisible et variée des dommages réels sur le verre. En appliquant diverses forces, nous avons pu produire un éventail de fractures et de rayures, allant de fines abrasions de surface à des fractures plus prononcées. Cette approche a été choisie pour reproduire de près les types de dommages que les surfaces en verre peuvent rencontrer dans des conditions réelles—tels que ceux causés par des impacts, des débris ou des facteurs de stress environnementaux—assurant ainsi la pertinence et le réalisme de notre configuration expérimentale. Ces motifs de dommages représentatifs nous permettent d'analyser plus efficacement l'influence des imperfections du verre sur la performance des capteurs et les algorithmes de détection d'objets.

Deux motifs de fracture différents et leurs images résultantes sont présentés dans la Fig. 11 et la Fig. 12. Nous tenons à noter que nous avons considérablement varié les focales des caméras pour comprendre comment les images apparaissent en mise au point proche et lointaine. Les résultats montrent que même des motifs de rayures mineures apparaissent dans l'image de sortie, tandis qu'un motif de multi-fractures beaucoup plus fort peut brouiller presque toute l'image. Cette expérience fournit l'intuition sur laquelle notre cadre de simulation et de visualisation est construit.

### Augmentation de l'AP pour les piétons dans KITTI

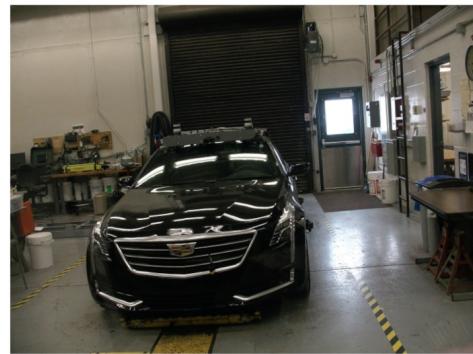
Nous tenons à souligner que l'augmentation de l'AP pour la classe des piétons était quelque chose qui nous a surpris au départ. Cependant, une analyse approfondie et qualitative nous a aidés à comprendre que cela se produisait en raison des fissures dans le verre, ce qui facilitait la classification des piétons par le modèle grâce à des contours améliorés autour d'eux. Ce n'était pas un artefact de contour, mais plutôt la fissure du verre agissant comme une bordure supplémentaire séparant clairement le piéton de l'arrière-plan. Un résultat similaire a également été observé dans [1] où l'AP global a été augmenté dans les images adversariales.



(a)



(b)



(c)

Figure 9 : Configuration expérimentale pour la collecte d'images affectées par des couches extérieures rayées/cassées pour une caméra. (a) montre l'ensemble de la configuration pour prendre des images adversariales. (b) montre la position de la caméra par rapport à la scène capturée. (c) montre la scène capturée par la caméra.



Figure 10 : Quelques motifs de fractures/éraflures sur le verre que nous avons utilisé pour collecter les images. (a) Une force vive appliquée perpendiculairement à la surface du verre, produisant des fractures se propageant radialement. (b) et (c) reproduisent un verre avec des éraflures

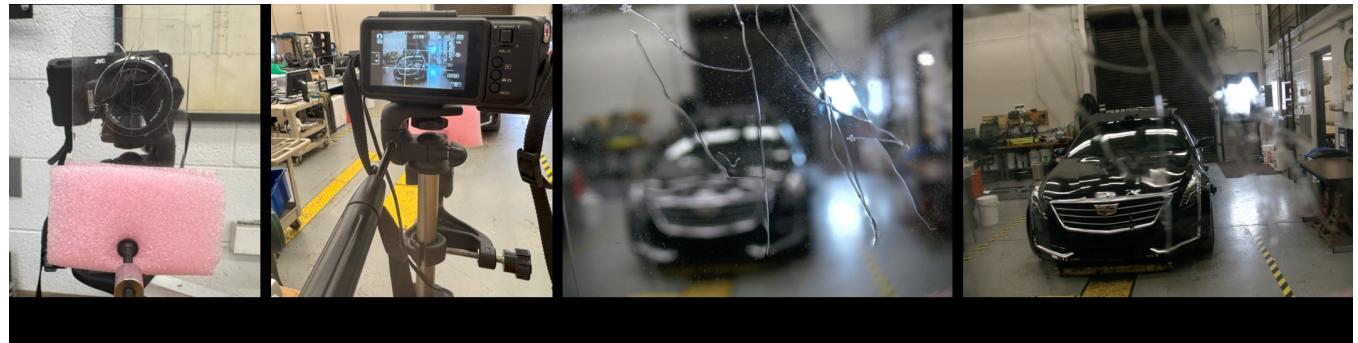


Figure 11 : (a) Montre le motif éraflé placé devant la caméra. (b) montre le point de vue de la caméra. (c) montre l'image capturée par la caméra (mise au point courte). (d) montre l'image capturée par la caméra (mise au point longue)

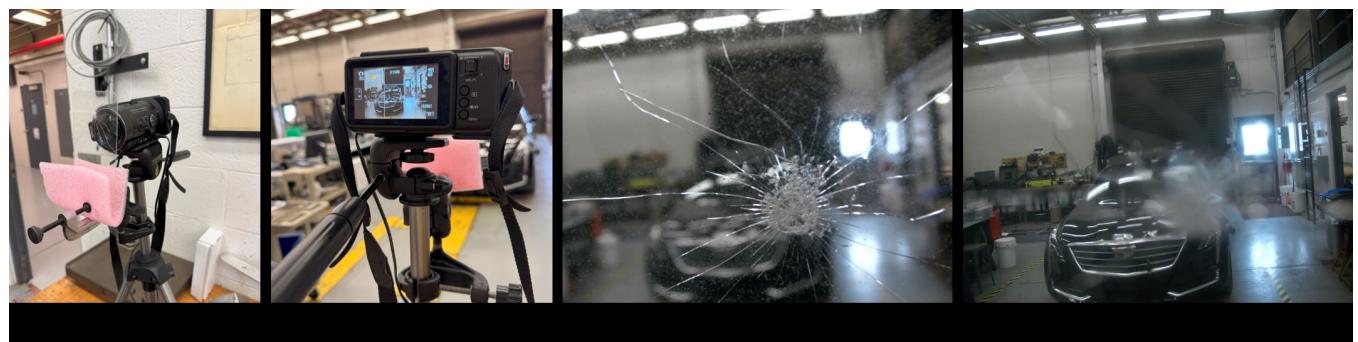


Figure 12 : (a) Montre le motif de verre brisé devant la caméra. (b) montre le point de vue de la caméra. (c) montre l'image capturée par la caméra (mise au point courte). (d) montre l'image capturée par la caméra (mise au point longue)

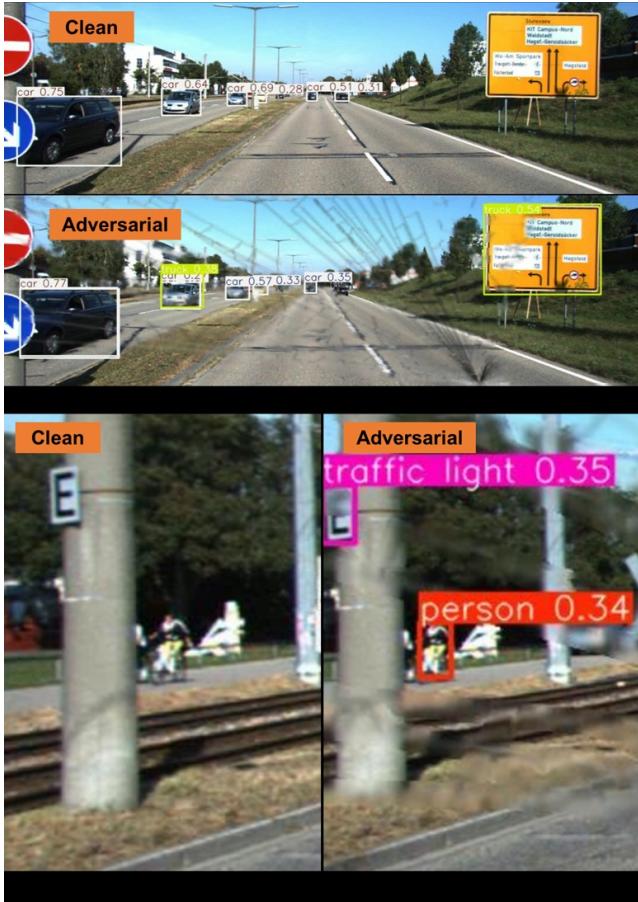


Figure 13 : (a) En haut - détections sur une image propre ; en bas - détections sur une image adversariable. (b) YoLo ne parvient pas à détecter la personne (c) Les fissures du verre permettent au modèle de détecter la personne.

### **Expérience Dynamique**

Dans cette section, nous décrivons l'expérience dynamique mentionnée dans la Section Introduction. Nous réalisons cette expérience pour comprendre la perturbation temporelle introduite par une fissure. Nous utilisons une fissure de pare-brise d'un véhicule et plaçons une petite caméra sur le tableau de bord derrière la fissure. Ensuite, nous photographions deux objets dynamiques - un véhicule et un piéton - alors qu'ils se déplacent dans la scène. La Fig. 14 fournit quelques images spécifiques avec l'inférence de YOLOv8 pour la classe véhicule. Nous montrons qu'avec la fissure, le véhicule reste indétecté dans la plupart des images. De plus, presque chaque image contient un faux positif. En conséquence, nous présentons la Fig. 15 comme les images avec une personne marchant dans la scène. Nous montrons qu'elle fournit par intermittence une détection et parfois avec une mauvaise classe (planche de surf).



Figure 14 : Images spécifiques des cadres pris avec la fissure du pare-brise avec l'inférence YOLOv8 pour la classe de véhicule. A - faux positif sans objet dans la scène ; B - pas d'inférence sur le véhicule ; C - pas d'inférence sur le véhicule ; D - première détection sur le véhicule ; E - deux détections différentes sur le même véhicule ; F - zone de délimitation incorrecte.

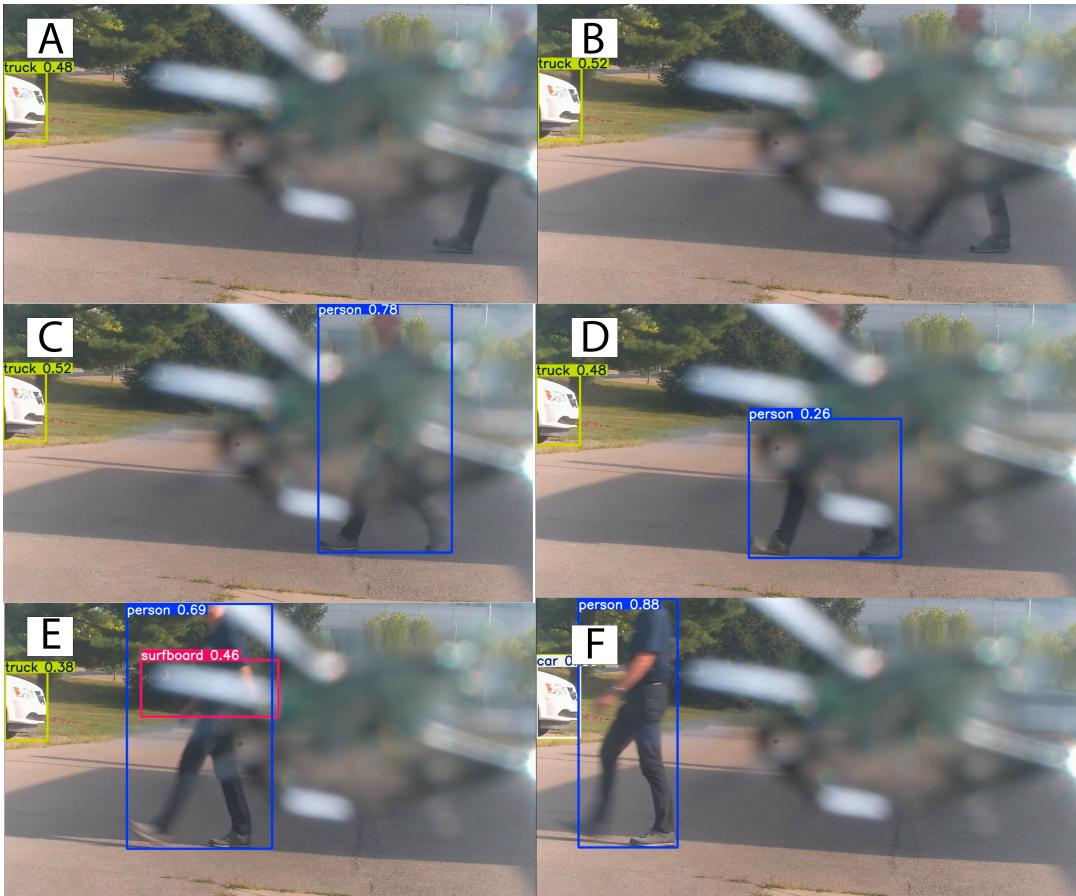


Figure 15 : Images spécifiques prises avec la fissure du pare-brise avec l'inférence YOLOv8 pour la classe personne. A - première entrée de la personne dans la scène sans détection ; B - aucune inférence de la personne ; C - première détection de la personne ; D - détection partielle de la personne ; E - détection de la personne avec une autre classe ; F - détection complète de la personne.

### **Images réelles de fractures de verre**

Nous présentons un exemple d'images de fractures de verre collectées sur le site FreePik, superposées sur le jeu de données KITTI avec l'inférence YOLOv8 (Fig. 16). Nous montrons que la fracture supprime certaines détections et diminue la confiance de détection d'autres.

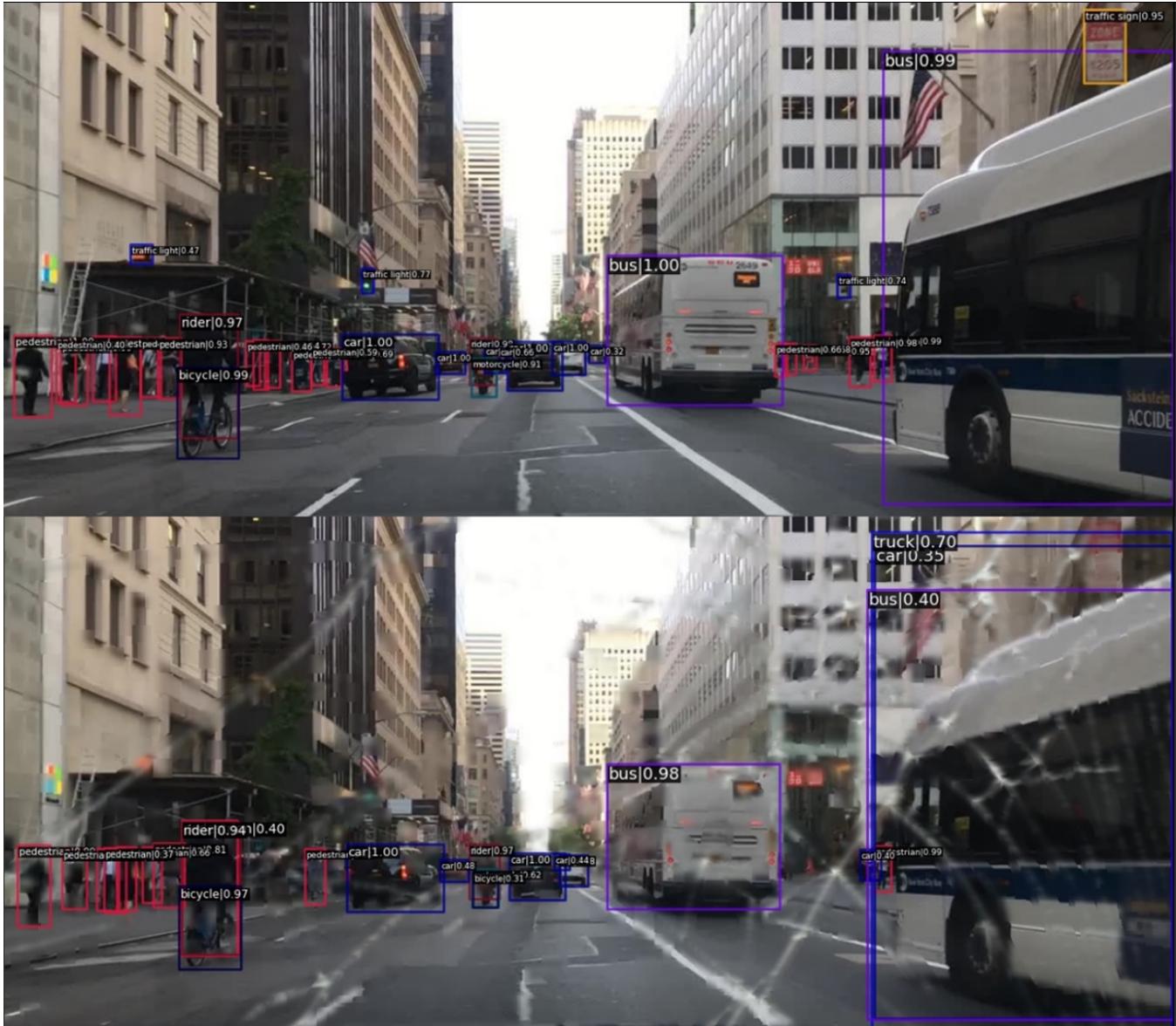


Figure16 : Haut - Inférence de PTv2 sur une image propre de BDD100k. Bas - Inférence pour une image réelle de verre brisé superposée sur BDD100k pour comparaison. Nous observons deux faux positifs supplémentaires sur le côté droit (camion, voiture) et plusieurs faux négatifs pour la classe piéton sur la gauche de l'image adversariale.