Dériver une base de données d'images en une nouvelle version plus cohérente et équilibrée pour de l'apprentissage profond

Cyril Barrelet, Marc Chaumont, Gérard Subsol, Vincent Creuze, Marc Gouttefarde

Dériver une base de données d'images en une nouvelle version plus cohérente et équilibrée pour de l'apprentissage profond

Cyril BARRELET[1], Marc CHAUMONT[1,3], Gérard SUBSOL[1], Vincent CREUZE[2], Marc GOUTTEFARDE[2]

[1]Equipe ICAR, LIRMM, Univ Montpellier, CNRS, Montpellier, France

[2]Equipe DEXTER, LIRMM, Univ Montpellier, CNRS, Montpellier, France

[3]Univ Nîmes, France

{cyril.barrelet, marc.chaumont, gerard.subsol, vincent.creuze, marc.gouttefarde}@lirmm.fr

Résumé – La multiplication des bases de données accessibles publiquement permet de couvrir de nombreuses applications du monde réel. Cependant, certains domaines sont encore très peu explorés, et les bases de données correspondantes sont moins cohérentes et plus sujettes aux erreurs du fait de leur utilisation restreinte. De plus, certains biais peuvent améliorer artificiellement les performances d'un modèle de Deep Learning (DL). Par conséquent, l'évaluation d'un modèle DL nécessite une compréhension claire de la base de données et une méthodologie adaptée pour le comparer à d'autres. Enfin, un modèle reflète souvent la performance sur une base de données. Celle-ci peut se détériorer si un décalage existe entre la base de données initiale et une base de données applicative réelle. Cet article propose de visiter les problèmes ci-dessus en dérivant une version plus cohérente et équilibrée de la base de données vidéo TrashCan de déchets sous-marins.

Abstract – The multiplication of publicly available databases makes it possible to cover many real-world applications. However, some domains are still very poorly explored, suggesting that databases are less consistent and more prone to errors because of their rare utilization. In addition, some biases can artificially improve the performance of a Deep Learning (DL) model. Therefore, evaluating a DL model requires a clear understanding of the database and a suitable methodology to compare it to others. Finally, a model often reflects the performance on a database. It can deteriorate if a shift exists between the initial database and a real-world applicative database. This paper proposes to visit the above problems by deriving a more consistent and balanced version of the TrashCan database of underwater macro-waste.

1 Introduction

L'augmentation de la pollution sous-marine par les macro-déchets [2] a récemment accru l'intérêt pour le nettoyage des fonds marins. La surveillance des macro-déchets déposés sur les fonds marins, notamment par acquisition optique, est devenu un sujet de recherche très actif [3]. En particulier, des approches de Deep Learning (DL) sont adaptées pour détecter, localiser et identifier les macro-déchets en environnement sous-marin. Ces approches peuvent fonctionner rapidement, ce qui est important si l'on veut guider des véhicules sous-marins télécommandés (ROV) pour ramasser ces déchets.

La disponibilité de base de données vidéos, telle que TrashCan [1], a permis aux chercheurs de concevoir des réseaux DL pour classer [4] [5] et localiser [6] [7] les macro-déchets dans des images sous-marines. Fig 1 montre une image issue de cette base de données.

TrashCan est une base de données sémantiquement segmentée regroupant 7 212 images extraites consécutivement à partir de 312 séquences vidéos prises par JAMSTEC, depuis 1982, dans la mer du Japon.

FIG. 1: *Image de la base de données TrashCan.*

Il s'agit d'une version améliorée de la base de données Trash-ICRA19 [9]. 16 à 22 catégories sont représentées selon la version, mais elles sont mal équilibrées en nombre d'images. Nous avons aussi constaté que de nombreuses annotations sont incorrectes, mal localisées ou manquantes. De plus, certaines métadonnées, telles que la profondeur, la date ou l'heure, sont directement superposées sur les images, ce qui peut introduire des artefacts dans le processus d'apprentissage.

Un autre problème se pose lors de l'évaluation. Comme

les 7 212 images de TrashCan sont extraites séquentiellement de séquences vidéo, Lorsque nous utilisons la méthode standard de validation croisée à *k*-blocs, deux images consécutives et donc très similaires peuvent faire partie d'un bloc d'entraînement et d'un bloc d'évaluation, ce qui peut augmenter artificiellement la performance globale.

En conclusion, il y a très peu de bases de données de macro-déchets sous-marins disponibles, et elles souffrent d'un fort déséquilibre entre les classes, de nombreuses erreurs d'annotation, et d'une cohérence temporelle qui rend de nombreuses images de la base de données très proches.

Dans ce travail, nous proposons de reprendre la base de données TrashCan, qui est la plus complète et la plus pertinente pour la détection et la localisation de macro-déchets, et d'en dériver une nouvelle version qui sera plus cohérente et plus équilibrée. Dans ce qui suit, nous décrivons notre méthode. Bien que la méthodologie soit centrée sur une base de données spécifique, elle pourrait être utilisée pour dériver d'autres bases de données d'images présentant les mêmes limites.

Le reste de l'article est organisé comme suit. La partie 2 décrit le processus de dérivation lui-même permettant d'obtenir une nouvelle base de données appelée UNO (Underwater Non-natural Object). La partie 3 décrit, en détail, une méthodologie pour obtenir des k-blocs garantissant que les images d'une même vidéo ne se trouvent que dans un seul ensemble de données d'entraînement ou d'évaluation. Dans la partie 4, le détecteur YOLOv5 [7] est utilisé pour comparer les performances en utilisant TrashCan et sa version dérivée UNO. Nous évaluons également les capacités de généralisation des deux modèles à l'aide d'une autre base de données, AquaLoc.

2 Construction de UNO

2.1 Redefinition des étiquettes

Bien que l'objectif de la base de données TrashCan soit de développer des méthodes de localisation de macro-déchets, elle comporte des catégories de non-déchets comme les éléments du ROV. Quant aux catégories macro-déchets, elles sont décomposées en 8 catégories distinctes avec 142 à 2 040 exemples par catégorie. Afin de pallier l'ambiguïté de la définition des déchets et à leur déséquilibre, nous avons choisi de fusionner les catégories *ROV* et déchets en une catégorie unique que nous avons nommée Non-Natural Object.

2.2 Suppression du texte

Comme le montre la figure 1, certaines métadonnées sont directement superposées dans les images sosu forme de texte. Nous avons choisi les supprimer car elles peuvent perturber le processus de détection dans la phase d'apprentissage ou d'évaluation car le texte peut être considéré comme un objet. Pour cela, nous avons découpé les images en haut et en bas. Nous avons utilisé un réseau de détection de texte poru définir automatiquement les dimensions de recadrage.

2.3 Relocalisation

TrashCan comporte une partie non négligeable d'annotations qui sont incorrectes voire manquantes, où dont les boîtes englobantes (BB, pour *Bounding Boxes*) correspondantes sont localisées de manière imprécise. Chaque BB a donc été relocalisée en assurant un meilleur centrage sur l'objet, tout en supprimant les BB incorrectes.

2.4 Discussion sur le jeu de données UNO

Certaines BB qui se chevauchent ont été incluses tout en minimisant leur nombre pour limiter la confusion au moment de l'évaluation. De plus, les objets fins en diagonale, tels que les cordes, peuvent perturber le modèle lors de l'apprentissage car ils créent des très grande BB ne contenant finalement que peu de contenu significatif.

3 Méthode d'équilibrage

Lorsqu'une base de données contient un petit nombre d'images, la meilleure façon d'évaluer correctement la précision d'un réseau de DL consiste à effectuer une validation croisée, c'est-à-dire de diviser la base de données en *k* blocs et à utiliser circulairement *(k-1)* blocs comme jeu de données d'apprentissage, et le dernier bloc comme jeu de données d'évaluation. La moyenne et l'écart-type des performances peuvent être ainsi calculés, permettant de donner un bon indicateur de généralisabilité et d'effectuer un test de Student pour une meilleure comparaison entre deux réseaux.

Dans le cas de TrashCan, une division aléatoire introduit un biais. Comme mentionné plus tôt, chaque image est extraite séquentiellement de plusieurs vidéos, de sorte qu'une image n et $n+1$ d'une vidéo peuvent être très similaires, ce qui conduit à l'éventualité d'avoir une image dans un bloc d'apprentissage et la suivante, quasi identique, dans le bloc d'évaluation. Par conséquent, la manière correcte de diviser l'ensemble de données est de regrouper toutes les images appartenant à une même vidéo dans un bloc unique. La figure 2 montre le nombre d'images par vidéo.

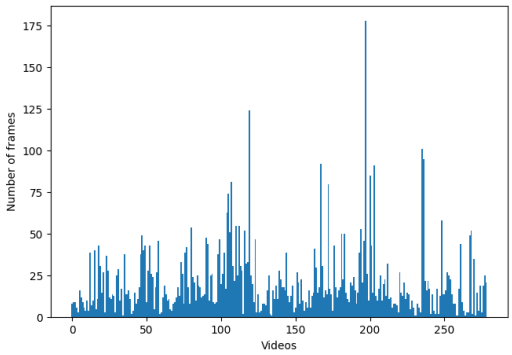

FIG. 2: *Nombre dimages pour les 279 vidéos.*

Nous devons donc maintenant diviser la base de données en k blocs (dans nos expériences, nous avons fixé k à 5), où le nombre d'images et aussi le nombre de BB sont équilibrés tout en conservant toutes les images d'une même vidéo dans un même bloc. Ce problème est appelé problème du bin-packing (problème d'empaquetage), puisque nous voulons remplir au mieux $k = 5$ blocs, sachant que leur capacité est approximativement le nombre total d'images divisé par k et approximativement le nombre total de BB divisé par k avec la contrainte des vidéos et le fait que nous cherchons à obtenir approximativement, pour chaque bloc, le même nombre d'images et le même nombre de boîtes englobantes.

Par conséquent, nous cherchons à minimiser à la fois l'écart type std_I du nombre d'images par bloc et l'écart type std_{BB} du nombre de BB. Comme le nombre de BB est similaire au nombre d'image (il y a de 0 à 3 BB par image en général), nous pouvons modéliser ce problème sous la forme :

$$f^* = \underset{\{f\}}{arg} \underset{f \in \{1..5\}^{279}}{min} \Big(std_I + std_{BB} \Big) \tag{1}$$

Avec f un 279-uplet, c'est-à-dire $f \in \{1,...,5\}^{279}$, correspondant à l'affectation de 279 vidéos à l'un des 5 blocs. Il y a 5^{279} différents 279 n-uplets, ce qui représente un nombre d'affectations possibles très grand.

Une solution approximative, mais rapide et facile, pour trouver la meilleure affectation f^* de ce problème consiste à créer 100 000 000 k-blocs en mélangeant les vidéos et en remplissant chaque bloc au mieux. Nous choisissons alors la répartition qui minimise l'équation (1). Nous obtenons std_I=3,01 et std_{BB}=9,60, ce qui semble être un résultat de variabilité très raisonnable pour des blocs contenant approximativement $7212/5 = 1442$ images et $10773/5 = 2154{,}6$ BB.

4 Résultats et expérimentations

4.1 Expériences

Nous avons choisi le modèle YOLOv5m [7] pré-entraîné sur ImageNet à une résolution de 640 × 640 pixels. Nous avons utilisé de l'apprentissage par transfert pour conserver les connaissances antérieures. Nous avons choisi l'optimiseur SGD et le planificateur OneCycle, avec un taux d'apprentissage initial et final de 0,0032 et 0,000384, respectivement, tout en fixant un échauffement à 20% du total des époques. De plus, nous avons fixé la taille des *batchs* à 28, ce qui est le maximum selon la capacité de notre GPU (NVIDIA Quadro RTX 6000 avec 24 Go de VRAM). Nous avons fixé le seuil de l'IoU à 0,2 entre les prédictions et les annotations pour l'évaluation. Nous avons également augmenté les données par transformation de couleurs, rotations, translations, mise à l'échelle, cisaillement, flip-UP, flip-LR, mosaïque et mixup. Comme mentionné, nous avons effectué cinq entraînements de 300 époques chacun et cinq tests pour chaque expérience. Chaque expérience a duré environ sept heures.

4.2 Résultats

4.2.1 Améliorations de UNO par rapport à TrashCAN

Le tableau 1 montre les résultats de diverses expériences avec les mêmes paramètres et le même *k*-blocs pour comparer les bases de données TrashCAN et UNO. Dans la première ligne, nous évaluons les résultats sans *k*-blocs, en divisant les ensembles aléatoirement pour montrer le biais lié aux bases de données contenant des images consécutives. Les résultats sont biaisés car les ensembles d'apprentissage et de validation contiennent des images similaires.

Dans la deuxième ligne, nous évaluons le modèle TrashCan à l'aide de notre méthodologie, tandis que dans la troisième ligne, nous évaluons le même modèle, mais sur l'ensemble de validation UNO pour voir ses performances sur une version plus cohérente. Ce dernier résultat montre une amélioration globale.

Enfin, nous donnons l'évaluation du modèle UNO sur son ensemble d'évaluation dans la quatrième ligne. En plus de voir ses résultats augmenter de presque 10 points, YOLOv5 a obtenu un écart-type plus faible, indiquant que UNO présente de meilleures propriétés de stabilité entre blocs.

4.2.2 Test de décalage covarié

150 images provenant de 3 vidéos prises dans le jeu de données AquaLoc en Méditerranée, en eau peu profonde, ont été annotées. La couleur, la luminosité, la turbidité, les objets et l'environnement étant très différents des images issues de TrashCan, la distribution des deux ensembles de données a peu de chance d'être similaires, et permet donc de tester la généralisation d'un modèle en testant le décalage covarié.

Le tableau 2 montre les résultats des deux modèles sur l'ensemble de données *décalé*. Bien que nous n'ayons pas trouvé de différence notable dans le score F1 des deux modèles, le modèle UNO obtient un score mAP@.5 plus élevé que TrashCAN. Nous avons supposé une amélioration réelle puisque nous avons corrélé les résultats mAP@.5 avec un test de Student et n'avons pas réussi à rejeter l'hypothèse nulle (signifiant que les deux modèles sont équivalents), avec une valeur p supérieure à 0,05 (p = 0,49).

Nous constatons des performances relativement faibles pour les modèles TrashCAN et UNO lors du test de décalage covarié. En effet, les images JAMSTEC et AquaLoc sont très différentes en termes de couleur et de forme des objets. De plus, un phénomène biologique appelé bio-encrassement marin, qui ne se forme pratiquement qu'en eau peu profonde, rend la tâche de détection plus difficile dans les images AquaLoc. Les images de JAMSTEC n'intègrent pas cette variabilité. Nous pourrions envisager de réaliser du *domain adaptation* [15], ou des méthodes génératives [16]. Ceci est envisagé dans des travaux futurs.

Entraînement	Evaluation	Répartition	P (%)	R (%)	F1 (%)	mAP@.5 (%)
TrashCAN	TrashCAN	Aléatoire	83.1	76.5	79.7	80.8
TrashCAN	TrashCAN	K-blocs	57.1 ± 5.2	60.1 ± 5.5	58.4 ± 4.2	56.6 ± 6.3
TrashCAN	UNO	K-blocs	64.2 ± 5.0	58.2 ± 2.9	60.9 ± 2.6	60.8 ± 4.2
UNO	**UNO**	**K-blocs**	**68.7 ± 4.3**	**66.2 ± 1.5**	**67.3 ± 1.5**	**68.8 ± 1.2**

TAB. 1: *Résultats sur différents ensembles de données obtenus en utilisant YOLOv5m*

Entraînement	Evaluation	Répartition	P (%)	R (%)	F1 (%)	mAP@.5 (%)
TrashCAN	AquaLoc	K-blocs	59.8 ± 6.3	52.9 ± 3.8	55.7 ± 1.6	52.5 ± 1.9
UNO	**AquaLoc**	**K-blocs**	**61.2 ± 3.1**	**51.2 ± 6.2**	**55.6 ± 4.5**	**55.2 ± 4.7**

TAB. 2: *Résultat du décalage covarié en utilisant YOLOv5m*

5 Conclusion

Dans cet article, nous avons d'abord présenté UNO, une version de TrashCan plus consistante et équilibrée, afin de créer une nouvelle base de données d'objets sous-marins non naturels.

Deuxièmement, nous avons proposé une méthodologie pour comparer les réseaux en utilisant un *k-fold* bien équilibré et avons conclu, grâce aux comparaisons de réseaux, que UNO présente de meilleures propriétés que TrashCan.

Troisièmement, nous avons évalué TrashCAN et UNO en utilisant YOLOv5 avec le même *k-fold* et les mêmes hyperparamètres pour une comparaison équitable.

Enfin, nous avons évalué l'efficacité de l'apprentissage dans des conditions de déploiement avec un test de décalage des covariables, en utilisant des images sous-marines tirées d'AQUALOC pour les modèles TrashCAN et UNO.

Comme mentionné, UNO ne contient qu'une seule catégorie en raison du déséquilibre des catégories de TrashCAN. Cependant, il est possible de surmonter ce problème en créant une base de données de déchets sous-marins bien équilibrée ou en utilisant des méthodes de classification *few-shot*: [13]. [14]. En outre, le test de décalage des covariables indique des performances de détection médiocres pour TrashCAN et UNO. Les images de JAMSTEC ne couvrant pas la variabilité du domaine cible, nous pourrions envisager de réaliser du *domain adaptation* [15] ou des méthodes génératives [16].

Références

[1] M. Goossens, F Mittelbach et A. Samarin. *TrashCan: A Semantically-Segmented Dataset towards Visual Detection of Marine Debris*. ArXiv, 2020

[2] Miquel Canals et al. *The quest for seafloor macrolitter: a critical review of background knowledge, current methods and future prospects*. Environmental Research Letters, 2020

[3] Madricardo Fantina et al. *How to Deal With Seafloor Marine Litter: An Overview of the State-of-the-Art and Future Perspectives*. Frontiers in Marine Science, 2020

[4] Wu Huiyan et al. *Multi-Level Feature Network With Multi-Loss for Person Re-Identification*. IEEE Access, 2019

[5] Tan Mingxing et Le Quoc. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proceedings of the 36th International Conference on Machine Learning, 2019

[6] Ren Shaoqing et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015

[7] Glenn Jocher et al. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. Zenodo, 2020

[8] Japan Agency for Marine Earth Science and Technology. *https://www.jamstec.go.jp/e/*.

[9] Fulton Michael et al. *Robotic Detection of Marine Litter Using Deep Visual Detection Models*. 2019 International Conference on Robotics and Automation (ICRA), 2019

[10] Maxime Ferrera et al. *AQUALOC: An underwater dataset for visual–inertial–pressure localization*. The International Journal of Robotics Research, 2019

[11] Lin Tsung-Yi et al. *Microsoft COCO: Common Objects in Context*. Computer Vision – ECCV 2014, 2014

[12] Petra Schwerin et Gerhard Wäscher. *The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP*. International Transactions in Operational Research, 1997

[13] Yuqing Hu et al. *Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning*. CoRR, 2021

[14] Yuqing Hu et al. *Sill-Net: Feature Augmentation with Separated Illumination Representation*. ArXiv, 2021

[15] Tzeng Eric et al. *Adversarial discriminative domain adaptation*. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017

[16] Jungseok Hong et al. *A Generative Approach Towards Improved Robotic Detection of Marine Litter*. 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020