

# Documentation des tunnels par imagerie 3D haute-résolution

■ Florian BARCET

*L'objectif de mon stage a été de mettre au point des méthodes d'apprentissage profond pour détecter précisément les équipements dans les tunnels à partir de données. Ma contribution intervient en premier lieu dans la constitution de la base de données labellisées à partir d'images en intensité et en profondeur issues du capteur LCMS. Pour cela, la modélisation de la forme du tunnel a été nécessaire avant d'utiliser des techniques de Machine Learning permettant la segmentation non supervisée des objets. Ensuite, des corrections manuelles ont été apportées pour obtenir des images segmentées et labellisées constituant nos vérités terrain.*

*En seconde partie, j'ai mis en œuvre et évalué une architecture d'apprentissage profond spécifiquement dédiée à la segmentation d'instances. Pour cela, j'ai utilisé des algorithmes existants tels que Mask R-CNN que j'ai adapté à ma problématique. J'ai réalisé plusieurs apprentissages grâce aux images d'intensité uniquement, puis j'ai fait varier les hyperparamètres et les bases de données en entrée.*

*Enfin, j'ai utilisé les résultats des modèles entraînés afin d'aller vers une documentation des équipements des tunnels. Pour cela, j'ai essayé de travailler non plus à l'échelle de l'image, mais à l'échelle du tunnel, en utilisant des techniques de fusion d'objets qui ont, elles aussi, été évaluées.*

## Contexte et objectifs de l'étude

La détection automatique d'équipements de tunnels est un projet que le CEREMA a débuté il y a plusieurs années. Ce projet a pour but de permettre la surveillance et le contrôle des ouvrages d'art constituant le patrimoine routier. L'objectif est de pouvoir réaliser une documentation détaillée des tunnels afin d'obtenir un suivi précis des éléments de sécurité.

Cette problématique a été abordée sous plusieurs angles. Premièrement, l'équipe de recherche ENDSUM, a développé un système d'acquisition de données innovant permettant d'obtenir des images précises des tunnels alliant intensité et profondeur. Puis, un travail de modélisation de la forme du tunnel a été entrepris avec des ajustements de surface. Cela a ouvert aux chercheurs l'approche du *Machine Learning* permettant de détecter les équipements. Cette partie a abouti au développement d'une interface utilisable pour l'annotation de données.

### MOTS-CLÉS

Deep Learning, Machine Learning, imagerie, tunnels, documentation

Dans ce contexte, l'objectif de mon stage a été de mettre au point des méthodes d'apprentissage profond pour détecter précisément les équipements dans les tunnels à partir de données. Ma contribution intervient notamment dans la constitution de la base de données labellisées, la mise en œuvre et l'évaluation d'une architecture spécifique dédiée à la segmentation d'instances, et l'utilisation des résultats pour aller vers une documentation des équipements des tunnels.



## Présentation et gestion des données

Les données utilisées sont issues du capteur développé par l'entreprise Pavemetrics appelé LCMS (*Laser Cracks Measurement System*). Son fonctionnement est fondé sur une technologie innovante : un émetteur projette une ligne laser sur l'objet inspecté, et une caméra digitale se charge d'enregistrer l'intensité de retour du signal ainsi que les déformations de la ligne (traduits en informations de profondeur). Ces profils sont capturés dans le sens du déplacement et sont agglomérés pour former des images de 2080 x 1500 pixels (figure 1). Ce système a une résolution fine : un point par millimètre le long de la ligne, un profil tous les 2 mm dans le sens d'acquisition et une précision de 0.5 mm en profondeur. Il a été conçu pour être utilisé sur des routes et autoroutes, mais nous l'avons adapté aux murs (parements) des tunnels. Le CEREMA a développé un prototype appelé MALT "*Mobile Acquisition with Laser in Tunnel*" (Foucher et al., 2019). Il a permis d'obtenir un grand nombre de données dans trois tunnels, utilisables pour les développements des programmes et interfaces.

Un des objectifs du projet est de modéliser l'aspect du tunnel. La problématique est que la forme d'un tunnel n'est pas une structure simple de type plan ou

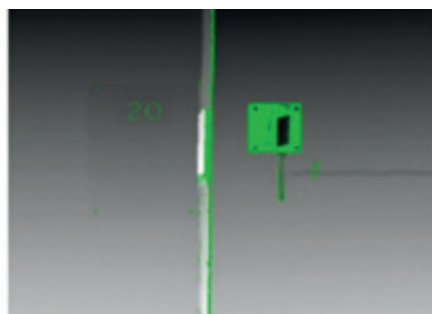


Figure 1. Exemple d'images d'intensité (à gauche en niveau de gris) et de profondeur (à droite) : niveau de gris selon la profondeur et valeurs invalides / hors limites sont marquées en vert (Tual et al., 2021).



cylindre. En effet, les murs sont souvent équipés d'éléments de sécurité (panneau de circulation, danger, incendie, etc.) ainsi que de différents câbles, gaines, lumières ou encore cavités, niches et refuges. En nous appuyant sur des méthodes d'estimation robuste, nous pouvons obtenir à la fois un modèle qui s'approche au mieux de la réalité et isoler les équipements, ce qui facilite la constitution de la vérité terrain nécessaire à l'apprentissage. Le choix de la fonction mathématique permettant de modéliser la surface est aussi très importante. Maxime TUAL (Tual et al., 2021) a choisi durant son stage au CEREMA d'utiliser des B-splines, car elles s'adaptent particulièrement bien à la forme d'un tunnel et à ses différentes caractéristiques. Afin de gérer l'implémentation de cet ajustement, un applicatif python a été développé. Cette interface permet de mettre en place une méthode incrémentale augmentant, à chaque itération, la robustesse. Les résultats obtenus avec l'ajustement de surface ont ouvert la voie à une approche de segmentation non supervisée sur des images considérant les résidus entre la surface ajustée et les points afin d'obtenir un masque au pixel près des objets. Cette segmentation s'est réalisée avec un algorithme hiérarchique appelé HDBSCAN (Campello et al., 2015). Différents hyperparamètres sont rajoutés afin d'accélérer la segmentation, ils ont été optimisés lors du projet de recherche technologique que j'ai réalisé au semestre précédent (BARCET, 2021). Pour finir, une interface a été implémentée afin de corriger les erreurs de segmentation et d'attribuer un

label à chaque classe. Cela nous permet d'obtenir des vérités terrain sur l'ensemble des images de tunnels (figure 2).

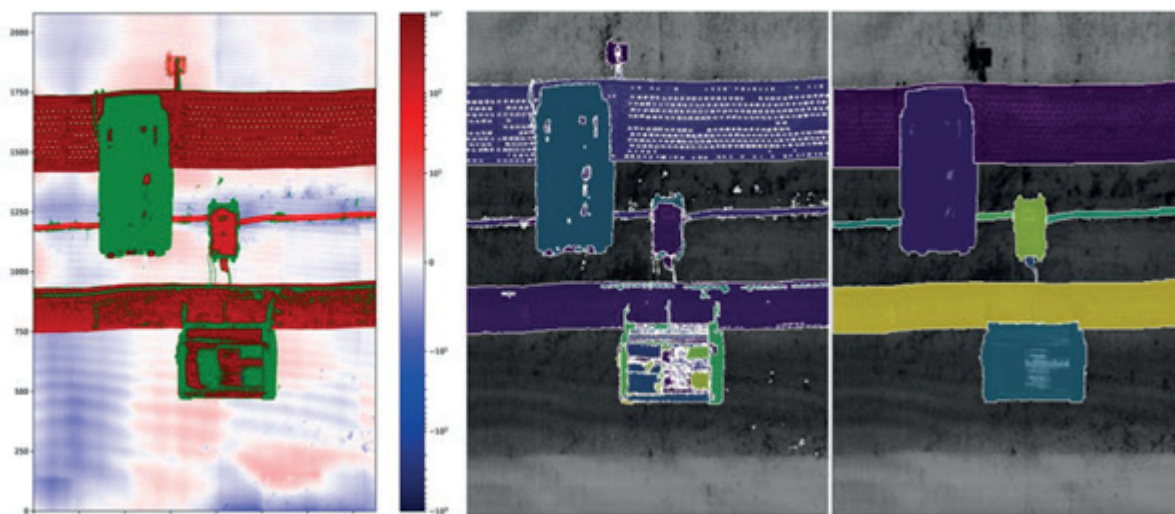
### Deep Learning et apprentissage

La reconnaissance des formes est une tâche qui se découpe en plusieurs catégories en fonction des besoins. On retrouve quatre grandes applications : la classification simple, la classification et la localisation, la détection d'objets et la segmentation d'instance. Depuis 2012, les algorithmes d'apprentissage profond (*Deep Learning*) sont devenus les plus performants pour ces tâches. Ils fonctionnent à partir de réseaux de neurones convolutifs organisés sous forme de couches. Celles-ci sont interconnectées (entièrement ou non), et elles reçoivent et interprètent les informations de la couche précédente. En tête de ces réseaux, nous retrouvons un extracteur de caractéristiques : une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de caractéristiques (*feature map*). Plus on réalise de convolutions, plus on va chercher des caractéristiques particulières dans une image. Au final, les cartes de convolution sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. C'est ce code qui est connecté à la seconde partie de l'algorithme, ayant pour but de combiner les caractéristiques afin de résoudre la tâche de reconnaissance des formes considérées.

Les objectifs du stage sont de détecter, à l'échelle du pixel, différents objets

complexes dans les tunnels. Pour cela, la segmentation d'instance est la plus appropriée. Le choix de notre algorithme s'est donc porté sur Mask R-CNN, qui est une référence dans ce domaine. Son architecture a été progressivement mise au point à travers une cascade d'articles publiés depuis 2014 avec l'architecture R-CNN (Girshick et al., 2014) dédié à la détection d'objets en introduisant des couches convolutives à certaines étapes de l'algorithme jusqu'à Faster R-CNN (Ren et al., 2016), une architecture entièrement convolutive avec l'introduction du réseau de propositions de régions (RPN). En 2018, l'architecture Mask R-CNN (He et al., 2018) ajoute l'extraction d'un masque binaire à l'algorithme Faster R-CNN, permettant une segmentation d'instance (figure 3).

Après un travail de création de vérités terrain, j'ai développé un programme de gestion de la base de données permettant de créer des jeux de données (datasets) d'entraînement, de validation et de test. Le premier sert à entraîner le réseau, le second à sélectionner le modèle et le dernier à tester et confirmer les résultats. J'ai ainsi pu réaliser plusieurs apprentissages sur un réseau préentraîné en faisant varier les effectifs et les données de chaque groupe. Pour introduire de la variabilité dans les données, les techniques usuelles d'augmentation de données ont été utilisées sur les datasets d'entraînement. Ces méthodes consistent à appliquer des transformations géométriques et photométriques sur les images d'apprentissage.



**Figure 2.**  
(Gauche) Résidus de l'ajustement robuste de la surface.  
(Centre) segmentation avec HDBSCAN.  
(Droite) Résultat après des opérations de post-traitement.

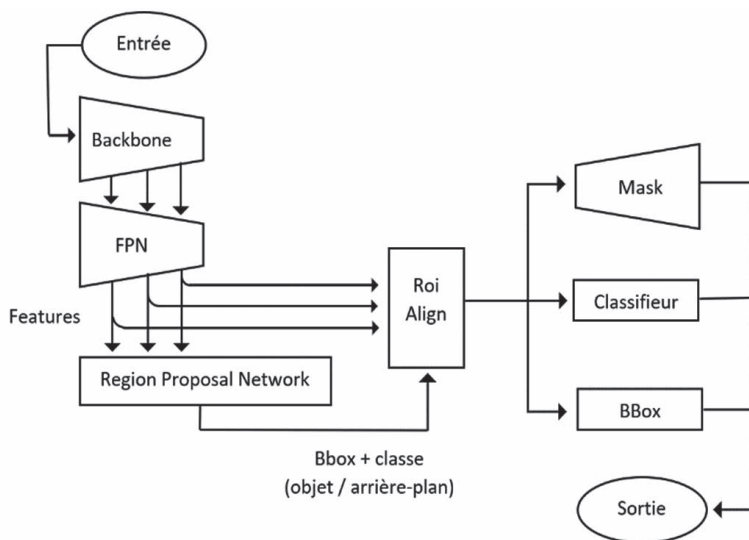


Figure 3. Diagramme des différents éléments du Mask-RCNN (Tual et al., 2021).

Dans les données tunnel en notre possession, 19 classes d'objets sont considérées. Notons que les effectifs entre les classes sont très déséquilibrés (nous avons, par exemple, 658 objets câbles différents et trois objets caméras). Il est difficile d'apprendre au réseau des équipements peu nombreux.

Le modèle est dans un premier temps évalué image par image, j'ai mis en place un algorithme capable de calculer, pour chaque classe, les objets bien détectés (vrais positifs, VP), les non détectés (vrais négatifs, VN), et ceux qui ont été détectés, mais qui n'auraient pas dû l'être (faux positifs, FP). Ces trois métriques vont nous conduire au calcul de la précision, qui va déterminer la qualité de détection de notre algorithme, et au rappel, qui montre l'exhaustivité de notre algorithme. La moyenne harmonique de ces deux outils donne le score F1 :

$$\text{Précision} = \frac{VP}{VP + FP} \quad \text{Rappel} = \frac{VP}{VP + FN}$$

$$F1\_score = 2 * \left( \frac{\text{Précision} * \text{Rappel}}{\text{Précision} + \text{Rappel}} \right)$$

Lors de l'évaluation des performances, on observe des résultats très encourageants (figure 4) sur les objets tels que les câbles, les passe-câbles et les luminaires, fortement représentés dans les données (Score F1 entre 80 % et 100 %). Pour les objets de fréquence d'apparition plus faible, les performances sont plus contrastées. Par exemple, les caméras de surveillance qui sont représentées par trois instances dans la base d'apprentissage sont très mal détectées. *A contrario*,

l'objet "prisme sos téléphone", également représenté par un faible nombre d'instances est très bien reconnu (score F1 de 100 %). On n'observe pas de corrélation entre le résultat de l'évaluation et le nombre d'objets représentés dans la base d'apprentissage. Une explication possible est que certains objets peu représentés ont une forme industrielle standardisée (panneaux de circulation par exemple) pour lesquels les caractéristiques varient peu et sont facilement apprises par l'algorithme. D'autres difficultés sont apparues, par exemple, les joints de voussoir sont particulièrement mal détectés. Cela semble être dû au fait que l'objet "joint" est en arrière-plan et qu'il est régulièrement recouvert par d'autres objets du premier plan.

Afin d'entraîner un modèle et évaluer ses performances quand on a des bases de données de petite taille, une des

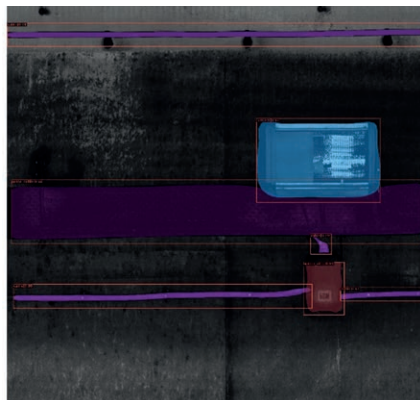


Figure 4. Exemple de résultat de détection sur une image comprenant un câble, un boîtier électrique, un passe-câbles et un luminaire.

solutions est la validation croisée. Elle se fonde sur des techniques d'échantillonnage. Elle permet de tester la stabilité de notre modèle et d'obtenir les meilleurs résultats possibles. Pour cela, nous divisons l'ensemble des données en n ensembles égaux. Le premier ensemble est sélectionné comme ensemble de validation et le modèle est entraîné sur les autres ensembles n-1. Le taux d'erreur est ensuite calculé après ajustement du modèle grâce aux données de validation. Dans la deuxième itération, le second ensemble est sélectionné comme validation et les autres ensembles n-1 sont utilisés pour l'entraînement. Ce processus se poursuit pour tous les n ensembles. J'ai donc adapté l'algorithme d'évaluation en utilisant la validation croisée sur l'ensemble des modèles obtenus. On obtient donc, pour chaque époque de chaque jeu de données, un fichier d'évaluation. J'ai ensuite comparé pour les mêmes époques les résultats afin de pouvoir calculer des moyennes et des écarts types. On observe que plus ce dernier est grand, moins la détection est stable. De plus, chaque fois que la détection dépasse une moyenne de score F1 supérieur à 75 %, l'écart type est inférieur à 20 %.

J'ai aussi pu voir que les objets les plus représentés étaient ceux ayant les résultats d'évaluation les plus stables (figure 5). Les écarts types pour les classes avec plus de 100 objets sont inférieurs à 5 %. Je rappelle qu'avoir un score stable ne veut pas dire avoir un bon score, l'algorithme trouve seulement de la même façon les objets, peu importe la configuration du jeu de données initial lors de l'apprentissage. Certains objets sont complètement mis de côté dans certains cas (caméras par exemple), cela est dû au fait que le tirage aléatoire utilisé ne garantit pas que les effectifs des classes suivent les mêmes proportions que celles de la population de départ. Pour contrer cela, un nombre minimum d'objets pourrait être indiqué dans chaque classe (technique de stratification). Cela permettrait d'obtenir des performances plus stables, mais cela implique aussi d'avoir un nombre minimum d'objets contenus dans nos classes. Il faut donc créer des vérités terrain supplémentaires pour ces objets.



	Précision		Rappel		F1		nb objets moyen
	Moyenne	Ecart type	Moyenne	Ecart-type	Moyenne	Ecart-type	
'non_classee'	0,20	0,45	0,07	0,15	0,10	0,22	33
'cable'	0,81	0,06	0,75	0,07	0,78	0,05	503
'passe_cable'	0,92	0,05	0,98	0,03	0,95	0,03	121
'lumiere'	0,92	0,06	1,00	0,00	0,96	0,03	84
'joint'	0,62	0,14	0,08	0,05	0,14	0,08	132
'camera'	0,00	0,00	0,00	0,00	0,00	0,00	1
'prisme_sos_telephone'	0,85	0,22	1,00	0,00	0,90	0,15	12
'bouche_incendie'	0,00	0,00	0,00	0,00	0,00	0,00	0
'reflecteur'	0,00	0,00	0,00	0,00	0,00	0,00	0
'prisme_issue_en_face'	0,40	0,55	0,40	0,55	0,40	0,55	2
'indication_issue_de_secours'	0,88	0,19	1,00	0,00	0,93	0,12	19
'plaque_numerotee'	0,60	0,55	0,60	0,55	0,60	0,55	4
'issue_de_secours'	0,20	0,30	0,40	0,55	0,26	0,37	4
'plaque_anneau'	0,00	0,00	0,00	0,00	0,00	0,00	0
'indication_id_sos'	0,77	0,23	1,00	0,00	0,85	0,15	10
'issue_sos_telephone'	0,83	0,18	1,00	0,00	0,90	0,11	19
'panneau_signalisation'	0,60	0,55	0,60	0,55	0,60	0,55	5
'coffrage'	0,50	0,50	0,60	0,55	0,53	0,51	3
'boitier_elec'	0,81	0,13	0,94	0,06	0,87	0,09	85

Figure 5. Résultat de la validation croisée pour la 140<sup>e</sup> époque.

## Fusion des objets

Un problème majeur est apparu lorsqu'on doit travailler avec des images "coupées" et qu'un objet se trouve au bord. Dans certains cas, il est difficilement détecté par l'algorithme, car les caractéristiques n'ont pas pu être suffisamment dévoilées. De plus, j'ai remarqué de nombreuses erreurs de classification pour ces objets. Ces problèmes étant liés au découpage (arbitraire) des données sous forme d'images. Il faut cependant distinguer deux types d'objets, les objets oblongs qui filent sur plusieurs images et les objets ponctuels. L'une des idées a été de reconstruire une image intermédiaire assurant le recouvrement de la zone. Les caractéristiques du capteur d'acquisition des données font que les images sont parfaitement contiguës les unes

avec les autres. Donc, pour créer les images de 50 % de recouvrement lors d'un déplacement de gauche à droite, il nous suffit de prendre la moitié droite de la première image et de la concaténer à la moitié gauche de la suivante.

Lorsque nous réalisons la détection sur les images se chevauchant, on obtient le plus souvent deux détections pour un même objet. Les objets qui ne sont détectés qu'une seule fois sont considérés comme des erreurs. Nous pouvons ensuite regrouper les détections d'un même objet (figure 6). Pour cela, nous passons les coordonnées de toutes les boîtes englobantes du référentiel image au référentiel tunnel. Puis, nous calculons l'IoU (*Intersection over Union*) d'un objet avec tous les autres de la même classe et nous gardons la valeur la plus élevée. Ensuite, pour choisir la meil-

leure détection, deux cas s'ouvrent à nous. Si l'objet est linéaire, alors toutes les détections sont conservées. Si l'objet est ponctuel, seules les détections qui ont été réalisées le plus au centre de l'image sont conservées. Cette méthode permet donc de détecter les objets longs aussi bien que les objets ponctuels et de les regrouper.

Pour évaluer la fusion, la première étape est de calculer un masque binaire par objet à partir des boîtes englobantes du groupe, ainsi, nous pouvons faire une mise en correspondance entre la vérité terrain et la détection. Ensuite, deux possibilités s'ouvrent à nous. D'un côté, nous pouvons définir un seuil à partir duquel le recouvrement n'est pas suffisant pour dire que l'objet est bien fusionné. De l'autre, nous pouvons compter le nombre d'objets

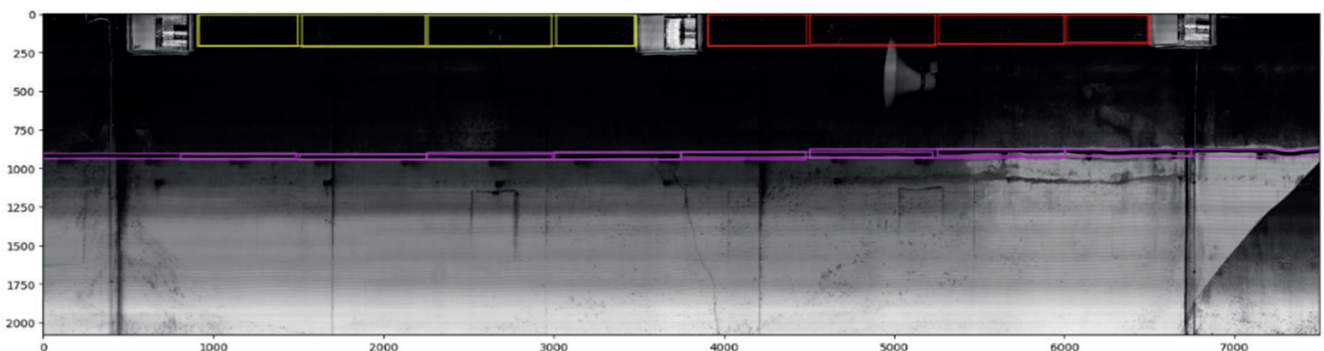


Figure 6. Résultat de la fusion de trois objets : câble (violet), passe-câbles (rouge et jaune).



dans chaque groupe. La détection doit avoir, dans son groupe, deux fois plus d'objets moins un que la vérité terrain. Nous pouvons ainsi calculer les mêmes métriques que lors de l'évaluation sur une image. Ainsi, j'obtiens une précision de 87 %, ce qui veut dire que les objets fusionnés le sont bien, cependant le rappel chute à 40 %. Le résultat du rappel n'est pas représentatif, car j'ai réalisé le calcul de façon globale (sur l'ensemble des classes), donc des objets peu représentés vont avoir un poids aussi important que les objets avec de forts effectifs dans le résultat du rappel. Ces objets peu représentés ont, en plus de cela, été mal appris par le détecteur. Cela implique que lors de l'évaluation de la fusion, ils vont fortement impacter le résultat, car ils ne seront pas trouvés. Deux solutions s'ouvrent alors à nous, la première est de faire le calcul de la précision et du rappel pour chaque classe, comme lors de l'évaluation de la détection. La seconde serait de pondérer le poids des scores en fonction de l'effectif afin de mettre moins en avant les petites classes.

Ce travail de fusion peu de ce fait encore être amélioré. Les principaux outils d'évaluations sont en place et vont pouvoir permettre des expérimentations sur de nouvelles données.

## Conclusion et perspectives

Durant mon stage, j'ai mis en œuvre et évalué une chaîne complète de détection d'équipement de tunnels dans le but d'obtenir une documentation détaillée de ces derniers. Ce projet de recherche s'inscrit dans la continuité de nombreux développements entrepris par le CEREMA. Ils étaient centrés sur des techniques de *Machine Learning* pour modéliser la forme des tunnels et créer des vérités terrain à partir de données d'intensité et de profondeur. Avec l'aide du projet de recherche technologique du semestre dernier, j'ai pu prendre en main ces outils, ce qui m'a permis de créer de façon optimisée des vérités terrain. J'ai donc pu constituer une base de données annotée de 24 classes avec environ 400 images d'apprentissage. J'ai adapté pour la problématique du stage l'architecture

Mask R-CNN dédiée à la segmentation d'instances. Afin de choisir ce type d'algorithme, j'ai dû réaliser un état de l'art sur l'ensemble de la cascade d'architectures existantes. Cela m'a permis de comprendre précisément le fonctionnement de cet algorithme ainsi que les hyperparamètres qu'il est possible de choisir et d'ajuster en fonction de la problématique.

Par la suite, j'ai développé des algorithmes de gestion de base de données afin de réaliser différents apprentissages. Avec le développement d'un algorithme d'évaluation classe par classe, j'ai pu optimiser les hyperparamètres et évaluer les performances sur plusieurs bases de test. Les résultats de ces expérimentations sont plutôt encourageants malgré un nombre réduit de données segmentées et annotées. On obtient de bons résultats sur les objets les plus représentés, comme les boîtiers électriques, les passe-câbles ou les luminaires. Un effort de création de vérité terrain pourrait être envisagé afin de consolider cette base, notamment pour les objets peu représentés.

Cette première partie m'a ensuite conduit sur des expérimentations à partir de modèles appris. J'ai pu réaliser les premiers tests pour la fusion des objets afin d'obtenir des résultats, non plus à l'échelle de l'image, mais à l'échelle du tunnel, afin d'aller vers une carte des équipements. Ce travail est rendu possible grâce à la nature même de notre système d'acquisition. Les images étant des profils agglomérés, nous pouvons reconstruire des images intermédiaires très facilement. Ensuite, des techniques de recouvrement vont nous permettre d'assembler les groupes d'objets ensemble, de supprimer les erreurs du détecteur et ainsi obtenir une documentation du tunnel. Ce travail obtient une précision intéressante, mais des tests plus poussés sur des nouvelles zones de tunnels peuvent encore être réalisés avec un modèle plus entraîné sur les objets rares.

En ouverture, plusieurs pistes peuvent encore être explorées. Premièrement, la modélisation de surface utilise encore des opérations manuelles qui pourraient être automa-

tisées en exploitant la connaissance des équipements issue des développements du stage. Le positionnement des nœuds des B-spline pourrait se faire sur les zones ne contenant pas d'équipement afin d'être plus précis. Cela nous permettrait d'obtenir de façon bien plus automatique les images constituées des résidus entre la surface ajustée et le nuage. Ce premier travail d'apprentissage sur les images d'intensité pourrait par la suite conduire sur la fusion de modalités. Cela permettrait d'ajouter la composante de profondeur à nos images et ainsi augmenter la performance de détection. Cela pourrait être très utile pour détecter des objets avec peu de caractéristiques saillantes dans les images d'intensité, comme les joints.

## Remerciements

Je voudrais tout d'abord adresser toute ma gratitude à M. Pierre Charbonnier, directeur du groupe ENDSUM, ainsi que M. Philippe Fouchier, chargé de recherche, pour leur confiance, et les connaissances qu'ils ont su partager avec moi. Je les remercie aussi pour leur disponibilité, la qualité de leur encadrement et l'autonomie qu'ils m'ont offerte pendant ce projet de recherche. Un grand merci à Maxime Tual qui m'a apporté son aide sur la programmation python ainsi que de précieuses indications sur le fonctionnement des programmes. ●

## Contact

Florian BARCET  
florian.barcet@outlook.com

## Références

Barcet, F. (2021). *Machine Learning pour le relevé d'équipements en Tunnel*. Rapport de projet de recherche technologique

Campello, R. J. G. B., Moulavi, D., Zimek, A., and Sander, J. (2015). *Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection*. *ACM Transactions on Knowledge Discovery from Data*, 10(1) :1-51.

Foucher, P., Charbonnier, P., Noël, T., Fosse, Y., and Hébert, J.-F. (2019). *Scanning tunnels with two very high-resolution laser devices and a stacker*. *The International*





Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W18 :39-46.

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

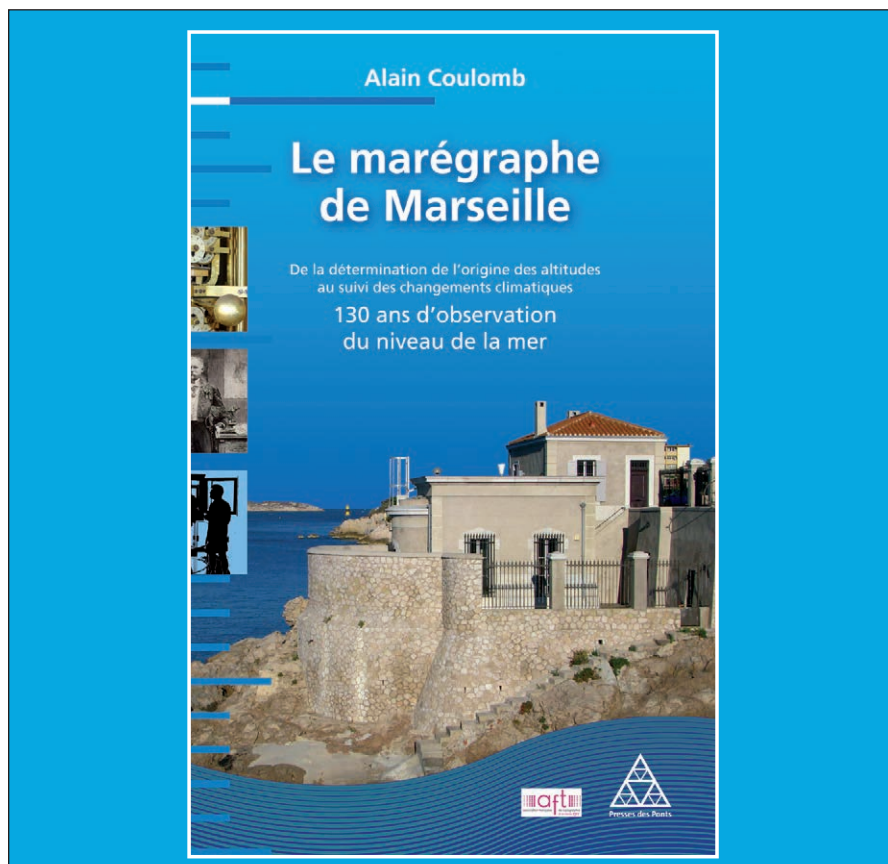
He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2018). *Mask R-CNN*. arXiv :1703.06870 [cs]. arXiv : 1703.06870.

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

Tual, M., Charbonnier, P., and Foucher, P. (2021). *Robust B-spline surface estimation for tunnel lining modelling and equipment surveying*. In NSG2021 2nd Conference on Geophysics for Infrastructure Planning, Monitoring and BIM, pages 1-5, Hybrid, France. European Association of Geoscientists & Engineers.

**ABSTRACT**

*The objective of my internship was to develop deep learning methods to accurately detect equipment in tunnels from data. My contribution comes first in the constitution of the labelled database from intensity and depth images from the LCMS sensor. For this purpose, modelling of the tunnel shape was necessary before using machine learning techniques allowing unsupervised segmentation of the objects. Then, manual corrections were made to obtain segmented and labelled images which constitute our ground truths database. In the second part, I implemented and evaluated a deep learning architecture specifically dedicated to instance segmentation. For this, I used existing algorithms such as Mask R-CNN that I adapted to my problem. I carried out several training sessions by varying the hyperparameters and the data. Finally, I used the results of the trained models in order to move towards a documentation of the tunnels' equipment. To do this, I tried to work not at the image scale, but at the tunnel scale by using object fusion techniques which were also evaluated.*



**COMMANDEZ "LE MARÉGRAPHE" au prix de 85,50 €**

640 pages, 400 illustrations frais de port inclus (France)

M/Mme Nom : \_\_\_\_\_

Prénom : \_\_\_\_\_

Société ou organisme : \_\_\_\_\_

Adresse : \_\_\_\_\_

Code postal : \_\_\_\_\_

Ville : \_\_\_\_\_

Tél. : \_\_\_\_\_

Fax : \_\_\_\_\_

Courriel : \_\_\_\_\_

Date : \_\_\_\_\_

Signature \_\_\_\_\_

**Bulletin de commande à retourner accompagné d'un chèque à l'Association francophone de topographie**  
73, avenue de Paris - 94165 SAINT-MANDÉ Cedex –  
Tél. : +33 (0) 1 43 98 84 80

**Achat également sur Internet : [www.aftopo.org](http://www.aftopo.org)**