



# **Rapport Final**

## **Analyse automatique d'images d'inspection immobilières**

**Rapport de stage**

**Léo BACKERT**

**Maîtrise en Informatique - Intelligence Artificielle**

Québec, Canada

© Léo BACKERT,

# Table des matières

<b>Table des matières</b>	ii
<b>Liste des tableaux</b>	iv
<b>Liste des figures</b>	v
<b>Remerciements</b>	vi
<b>1 Introduction</b>	1
<b>2 Littérature</b>	3
2.1 Détection d'objet dans une image . . . . .	3
2.2 Modèles de détection d'objet . . . . .	3
2.3 Données débalancés et augmentations de données . . . . .	4
2.4 Transfert d'apprentissage . . . . .	5
2.5 État de l'art . . . . .	5
<b>3 Données</b>	7
3.1 Jeux d'entraînement et de validation . . . . .	7
3.2 Jeu de test . . . . .	9
3.3 Divisions des jeux de données . . . . .	10
<b>4 Baselines</b>	12
4.1 Modèles . . . . .	12
4.2 Méthode . . . . .	13
4.3 Métriques . . . . .	15
4.4 Résultats de la baseline . . . . .	15
<b>5 Expérimentation</b>	17
5.1 Synthèse des résultats . . . . .	17
5.2 Analyse et discussion des résultats . . . . .	18
5.3 Recherche d'hyperparamètre . . . . .	19
5.4 Résultats tests . . . . .	20
5.5 Ouverture . . . . .	23
<b>6 Conclusion</b>	25
<b>Bibliographie</b>	26

<b>A</b>	<b>Architecture</b>	<b>28</b>
<b>B</b>	<b>Liste des classes hiérarchisées</b>	<b>29</b>
<b>C</b>	<b>Correspondances des classes en fonction de leurs code unformat II</b>	<b>33</b>
<b>D</b>	<b>Distributions</b>	<b>34</b>

# Liste des tableaux

4.1 Résultats baseline général . . . . .	16
4.2 Résultats baseline par classes . . . . .	16
5.1 Résultats expérience générale . . . . .	17
5.2 Résultats expérience par classes . . . . .	18
5.3 Synthèse du résultat de la recherche d'hyperparamètre sur le modèle R50_COCO	20
5.4 Matrices de confusion générale pour le modèle R50_COCO . . . . .	20
5.5 Matrices de confusion générale pour le modèle R101_COCO . . . . .	20
5.6 Matrices de confusion générale pour le modèle R50_CITY . . . . .	21
5.7 Résultats F $\beta$ -Scores . . . . .	21
5.8 mAP de la classe fenêtre en utilisant R50_COCO . . . . .	23
5.9 mAP de fenêtres, garde-corps et portes en utilisant R50_COCO . . . . .	23
5.10 AP par classes pour les 3 classes . . . . .	23
B.1 Listes des classes . . . . .	29
C.1 Descriptions des classes . . . . .	33

# Liste des figures

3.1	Exemples de fiftyone d'images annotées . . . . .	7
3.2	Exemple d'une image du jeu annoté automatiquement . . . . .	8
3.3	Exemples de données erronées dû à l'annotation automatique Les annotations sont en code Uniformat II (Annexe C). . . . .	8
3.4	Pipeline . . . . .	10
3.5	Pipeline . . . . .	10
4.1	Pipeline . . . . .	14
5.1	Exemples d'inférences . . . . .	22
A.1	Architecture du modèle . . . . .	28
D.1	Distribution des données nettoyées du jeu d'entraînement . . . . .	34
D.2	Distribution des données nettoyées du jeu de validation . . . . .	35

# Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès du stage en intelligence artificielle et qui m'ont aidé lors de la rédaction de ce rapport.

Je tiens aussi à remercier mon maître de stage M. Martin BLOUIN, expert intelligence artificielle, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien.

Je remercie également l'équipe de Hoodi pour leur accueil, leur esprit d'équipe pour leur accueil ainsi que l'opportunité qu'ils m'ont donnée de me confronter au monde de la recherche ainsi qu'à obtenir de nouvelles connaissances en imagerie notamment en détection d'objet. De plus, en tant qu'étudiant étranger, cette expérience permet d'être confronté au monde du travail québécois. Ces quatre premiers mois, me permettent aussi de me questionner sur mon projet professionnel. En effet, partir étudier ou travailler dans un autre pays demande de prendre en considération dans son projet professionnel sa vie personnelle comme ses amis et sa famille. Ce stage a été pour moi une expérience enrichissante autant personnellement que professionnellement.

Finalement, je tiens aussi à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage ainsi que des précédents : ma famille et mes amis en particulier Morgane ANDANSON camarade de maîtrise et M. Gabriel LECLERC.

# Chapitre 1

## Introduction

Durant la session d'automne 2022 et d'hiver 2023, j'ai effectué le stage en intelligence artificielle (IFT- 6010 et IFT-6501) chez Hoodi.ai à Québec. Au cours de cette première étape, j'ai pu m'intéresser à l'analyse automatique d'images en inspection immobilières. Mon encadrant pour ce projet est Ing. PhD. Martin BLOUIN. Chez Hoodi, il occupe la place d'associé, expert en intelligence artificielle. Dans ce projet, je peux donc apprendre dans d'excellentes conditions et bénéficier de soutiens de qualités.

Ensuite, ce projet me permet aussi d'en apprendre plus sur les enjeux en bâtiment et plus précisément sur les copropriétés du Québec. En effet, il existe une loi récente (La loi 16 pour les propriétaires de condos et les administrateurs de copropriétés divisées.), qui demande aux syndicats de copropriétés de faire des études de fonds de prévoyance ainsi que de tenir à jour un carnet d'entretien. Cependant, pour faire une étude de fonds, il faut faire appel à un professionnel du bâtiment. Mais malheureusement, ouvrir une étude et faire déplacer un professionnel à un coût élevé cela peut poser un problème pour des petites copropriétés (2 à 20 unités). En effet, le coût d'une étude de fonds n'est pas proportionnel en termes de ratio coût/propriétaire.

C'est dans cette optique-ci que Hoodi.ai propose différentes solutions numériques qui facilitent le maintien et l'entretien des copropriétés. Les deux produits vendus par l'entreprise que nous présenterons brièvement plus bas afin de déterminer le cadre du projet sont : le « Carnet d'entretien » et « Mon étude de fonds ».

## Carnet d'entretien

Le carnet d'entretien a pour cible les syndicats des copropriétés de toutes tailles. Il a pour but d'établir un registre dans lequel les papiers administratifs sont listés et sauvegardés (Plans, factures, etc.).

## Étude de fonds

Le second produit vendu par Hoodi.ai est « Étude de fonds ». Ce produit a pour cible les petites copropriétés et est la solution majeure proposée aux petites copropriétés en réponse à la loi 16. En effet, leur solution digitale réduit considérablement le prix d'une étude de fonds de prévoyance grâce à de nombreuses étapes effectuées numériquement sur le site de l'entreprise. Un des facteurs marquant de l'étude de fonds de Hoodi qui permet de réduire le coût est la visite virtuelle (téléversement des photos par le client).

En outre, il doit aussi établir la liste des composantes du bâtiment. Afin que le professionnel puisse évaluer l'état de ces composantes. Cette étape est actuellement manuelle et demande un peu de connaissance quant aux termes techniques et peut être difficile pour le client. C'est dans cette optique que l'entreprise a pensé simplifier cette étape demandée aux clients. En effet, l'entreprise aimeraient trouver une solution utilisant un algorithme d'apprentissage automatique (Machine Learning).

Cette solution est divisée en trois grands objectifs.

Premièrement, l'algorithme d'apprentissage machine devra détecter différentes composantes dans les photos prises par le client. De manière à pouvoir lister le plus exhaustivement possible les différents éléments.

Deuxièmement, l'algorithme, de la même manière que le premier objectif, devra permettre aux clients d'obtenir la liste du premier objectif, mais cette fois-ci avec les matériaux associés à leur composante.

Enfin, si le temps le permet, l'objectif est d'inclure un retour aux clients. C'est-à-dire, de lui communiquer des informations concernant ses images. Par exemple : le manque de composantes dans ses images ou de la qualité de ses photos (nettété de la photo).

Afin de répondre aux trois objectifs expliqués ci-dessus, le rapport sera divisé en plusieurs chapitres. Tout d'abord, nous présenterons la littérature effectuée pour ce projet. Ensuite, il serait intéressant de présenter en premier lieu les données et en second lieu, étudier les modèles et algorithmes utilisés pendant ce stage. Enfin, nous nous intéresserons aux expérimentations ainsi qu'aux résultats obtenus.

# Chapitre 2

## Littérature

Afin d'améliorer la compréhension de ce chapitre, nous allons structurer la littérature en différentes sections pour souligner les points importants du stage. Le problème de ce projet est un problème de détection d'objets dans une image en apprentissage supervisé. Nous nous intéresserons donc tout d'abord à la détection d'objet dans une image et au modèle de détection d'objet. Ensuite, nous étudierons les problématiques liées aux données du projet avec les manques et le débancement des données. Enfin, nous nous pencherons sur les travaux déjà existants dans la littérature dans le domaine de la détection des bâtiments d'habitations.

### 2.1 Détection d'objet dans une image

Tout d'abord, pour résoudre les objectifs du stage listés dans le chapitre 1, on commence par rechercher différents modèles permettant de mener à bien ces objectifs de détections d'objet (Bharati and Pramanik, 2020). De plus, les différents modèles du chapitre 5 seront comparés de la même manière que dans l'article. C'est-à-dire, la comparaison des modèles avec le même jeu de pré-entraînement, la même résolution d'images, etc. Cependant, seulement la précision sera retenue.

### 2.2 Modèles de détection d'objet

Les modèles de détections d'objet de la revue Bharati and Pramanik (2020) qui seront utilisés pour ce projet sont donc l'algorithme YOLO ainsi que l'algorithme Mask R-CNN. Ces deux algorithmes sont basés sur des réseaux de neurones à convolutions (CNN).

Le premier algorithme ainsi retenu pour effectuer la recherche de baseline est l'algorithme You only look once (YOLO) (Handalage and Kuganandamurthy, 2021). Comme son nom l'indique, il ne regarde les images qu'une seule fois. Cette caractéristique lui permet ainsi d'entraîner son réseau et d'inférer sur les images rapidement. Cependant, il demande de nombreuses instances par classe. En outre, il demande aussi une grande variété d'images d'après ultralytics (2022). Brièvement, cet algorithme va diviser l'image en grille et pour chaque case de la grille, YOLO prédit une boite englobante (bounding box) et donne aussi la probabilité de chacune des classes.

Au contraire, l'algorithme Mask R-CNN est plus lent, car il passe plus de temps à s'entraîner sur les images que la méthode précédente (He et al., 2017). Son algorithme est très similaire à celui du Faster R-CNN (Ren et al., 2015). Cependant, Mask R-CNN rajoute un masque à l'image qui, en plus d'utiliser un cadre pour trouver la région de l'objet, cherche aussi sa forme. Cela permet d'affiner la prédiction du modèle en utilisant la forme de l'objet.

Ensuite, le modèle masque R-CNN permettra de garder la même méthodologie entre les différents modèles identifiés dans le chapitre 5. Plus précisément, les modèles pré-entraînés utilisé sont tiré du framework detectron2 (Wu et al., 2019a). Avec les backbones suivants : ResNet-50 et ResNet-100 décrit dans l'article de ResNet He et al. (2015). En ajoutant une structure pyramidale (FPN) par-dessus décrite dans la documentation des modèles zoo donnés par detectron2 (Wu et al., 2019b) (Annexe A).

## 2.3 Données débalancés et augmentations de données

Troisièmement, ce projet fait face à un débancement de données. Pour régler ce problème, la revue Johnson and Khoshgoftaar (2019) propose différentes solutions. Comme le regroupement de classes, le rééchantillonnage des classes ou encore l'ajustement de la fonction de loss.

Pour le regroupement des classes, cela est réalisé à l'aide d'une hiérarchie en deux niveaux (Annexe B). L'échantillonnage, consiste simplement à rajouter naïvement un échantillon des annotations pour les catégories faiblement représenté dans le jeu ou inversement supprimer des annotations pour les catégories fortement représentées dans le jeu. Ensuite, l'ajustement de la fonction de loss on utilisera la focal loss décrite dans la revue. Cette méthode consiste à modifier la fonction de pertes qu'utilisent les modèles pour converger afin de pénaliser plus fortement les mauvaises prédictions lorsque le nombre d'annotations d'une classe est plus rare.

Enfin, une autre méthode qui permet de combattre le manque de données et l'augmentation de données. Cette méthode consiste à utiliser les données d'entraînement en leur appliquant des transformations afin de créer de nouvelles données artificiellement et ainsi permettre d'avoir plus de données.

## 2.4 Transfert d'apprentissage

Les données pour ce projet sont relativement limitées, il faudra donc faire appel à une technique de transfert d'apprentissage. Cette méthode consiste à entraîner un modèle sur un jeu de données avec des caractéristiques similaires aux classes de notre projet afin de pouvoir réutiliser ce modèle pour adapter ses caractéristiques apprises aux classes du projet, afin qu'elles répondent à nos objectifs. Les jeux de données ont été choisis pour leurs images qui sont prises dans leurs environnements naturels et pour leurs grandes quantités d'images extérieures.

Pour le transfert d'apprentissage, on utilisera donc des modèles pré-entraînés sur des jeux de données. Comme le jeu de donnée COCO qui contient de nombreuses classes d'éléments extérieurs dans des paysages urbains sur le continent d'Amérique du Nord (Lin et al., 2014) Cependant, pour maximiser nos chances de réussite, il faudrait utiliser des modèles pré-entraînés sur des données avec des objets similaires à notre problème. Sur Hugging Face (Face, 2022), il existe déjà quelques modèles pré-entraînés sur Cityscapes (Cordts et al., 2016). C'est un jeu de données contenant des images dans des milieux plus urbanisé que le jeu de COCO. Cependant, les images étant prises dans des villes allemandes, il faut rester attentif à l'architecture, car les bâtiments européens sont différents de ceux de la province de Québec car les études de fonds de Hoodi.ai sont réalisées au Québec (majoritairement dans les villes de Québec et de Montréal). De plus, Hugging Face propose aussi des modèles entraînés sur le jeu de données ADE20K (Zhou et al., 2016). Ces données contiennent beaucoup de classes du projet se trouvant à l'extérieur comme des maisons ou des catégories du bâtiment : des fenêtres, des portes ou escalier. Malheureusement, les données de ce jeu étant étiquetées de manière panoptique, alors que les données du projet sont étiquetées pour de la segmentation, il faudrait convertir les étiquettes des jeux de projets avant de pouvoir faire du transfert d'apprentissage. Cette tâche étant trop chronophage dans le contexte du stage, du transfert d'apprentissage sur ce jeu restera une ouverture pour le futur du projet.

## 2.5 État de l'art

Finalement, afin de placer au mieux possible le projet dans la littérature existante, on s'intéresse à des travaux déjà réalisés par la communauté scientifique sur la détection de composantes d'habitations comme les travaux réalisés sur le OneFormer (Jain et al., 2023). Le OneFormer est une architecture de modèle basée sur les transformers. Dans le contexte de ce rapport, on utilisera le modèle entraîné sur ADE20K (Zhou et al., 2016) dans le chapitre suivant sur les données (Chapitre 3).

De plus, dans le domaine de la reconnaissance d'habitation, il existe de nombreux autres travaux notamment dans la surveillance de l'état des structures dans les articles de Liu et al. (2021) et de Agyemang et al. (2021). Ces travaux sont très proches de l'objectif du projet dans le cadre d'une étude de fonds de prévoyance. Cependant, les travaux réalisés ne vont pas suffisamment profondément dans les détails des composantes. Plus précisément, ce projet ne doit pas seulement être capable de déterminer la composante (Par exemple : les fenêtres), mais la gamme de composantes dans laquelle elles appartiennent (Fenêtres : coulissante, en PVC, ouverture en battant, etc.) pour pouvoir connaître sa gamme de prix d'entretien correspondante.

# Chapitre 3

## Données

### 3.1 Jeux d'entraînement et de validation

Tout d'abord, en ce qui concerne les données du stage, pour avoir les données d'entraînement et de validation les plus représentatives de la réalité possible, il a été décidé de fusionner deux jeux de données différents.

#### 3.1.1 Données annotées manuellement



(a) Exemple d'images de façades



(b) Exemple d'images de photo similaire à ce qu'un client pourrait téléverser

FIGURE 3.1 – Exemples de fiftyone d'images annotées

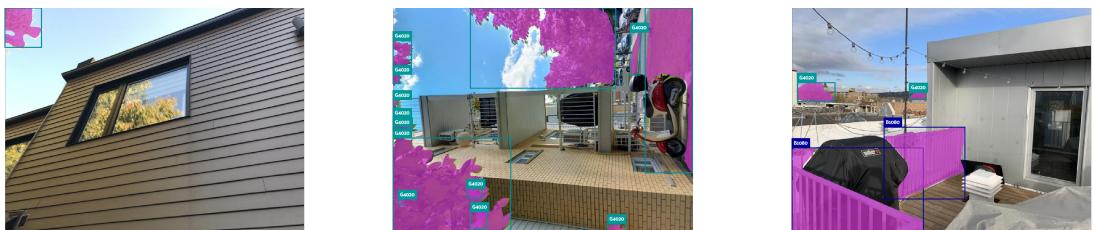
Le premier jeu de données (Figures 3.1) correspond à des données de façade de bâtiment prise dans le vieux Québec (Figure 3.1a) ainsi qu'à des données similaires à ceux que des clients pourraient poster lors de leurs démarches d'étude de fonds (Figure 3.1b). Ces données ont été étiquetées manuellement par des techniciens et ingénieurs dans le bâtiment chez Hoodi.ai. Cela donne un total de 335 images avec 2485 instances réparties dans 14 catégories. Ce jeu de données, malgré l'expertise des annotateurs, comporte tout de même quelques imperfections.

En premier lieu, ce jeu comporte peu d'images et est très débalancé le nombre élevé d'instances peut s'expliquer par les photos de façades qui elles comportent beaucoup de composantes (plusieurs fenêtres, porte, balcons, etc). Tandis que, les photos qui représentent ce que les clients pourraient soumettre eux ne comportent qu'une seule classe, car la photo a été prise pour montrer un élément spécifique. En second lieu, les éléments hors de la propriété n'ont pas été annotés, car dans le contexte d'une étude de fonds, la liste de composantes fournie par le modèle doit être dans la copropriété. Cependant, une fenêtre reste une fenêtre et cela pourrait biaiser l'apprentissage des modèles.

### 3.1.2 Données annotées automatiquement



FIGURE 3.2 – Exemple d'une image du jeu annoté automatiquement



(a) Annotation manquante

(b) Annotation fragmenté

(c) Annotations hors propriété

FIGURE 3.3 – Exemples de données erronées dû à l'annotation automatique  
Les annotations sont en code Uniformat II (Annexe C).

Le second jeu de données (Figure 3.2), quant à lui a pour but d'ajouter des données au projet. Il a été annoté automatiquement par le OneFormer entraîné sur ADE20K. Les images utilisées pour l'annotation automatique ont directement été récupérées d'anciennes études de l'entreprise et représentent donc exactement ce qu'un client soumettra sur le site. Ces données représentent donc un total de 1852 images avec 6 753 instances réparties dans 15 catégories différentes. Malheureusement, l'étiquetage automatique présente quelques limites (Figures 3.3). Bien que le modèle utilisé pour annoter automatiquement soit très bon, il n'est pas parfait et échoue sur quelques notations. (Figure 3.3a). De plus, cela crée des étiquettes imparfaites qui ne détourment pas aussi bien la composante qu'un humain pourrait le faire ou segmente une composante en plusieurs annotations (Figure 3.3b). Ensuite, contrairement au premier jeu, l'annotation automatique crée des annotations hors propriété (Figure 3.3c). Finalement, les catégories choisies pour être annotées automatiquement n'ont pas été sélectionnées par un expert du bâtiment et cela pourrait être dérangeant si la définition de la composante a été mal comprise.

Ensuite, pour les données de test, la structure des données sera très différente car l'objectif est de dresser un inventaire de composantes et non de placer les composantes dans l'image. C'est donc dans cette optique que le jeu de données de test a été construit. Il comporte au total 18 anciennes études de l'entreprise où chacune d'entre elle comporte une liste d'images (images téléchargées par le client) ainsi que la liste de composantes récupérées directement du rapport d'études de fond.

## 3.2 Jeu de test

Ensuite, pour les données de test, la structure des données sera très différente car l'objectif est de dresser un inventaire de composantes et non de placer les composantes dans l'image. C'est donc dans cette optique que le jeu de données de test a été construit. Il comporte au total 18 anciennes études de l'entreprise où chacune d'entre elle comporte une liste d'images (images téléchargées par le client) ainsi que la liste de composantes récupérée directement du rapport d'études de fond.

### 3.3 Divisions des jeux de données

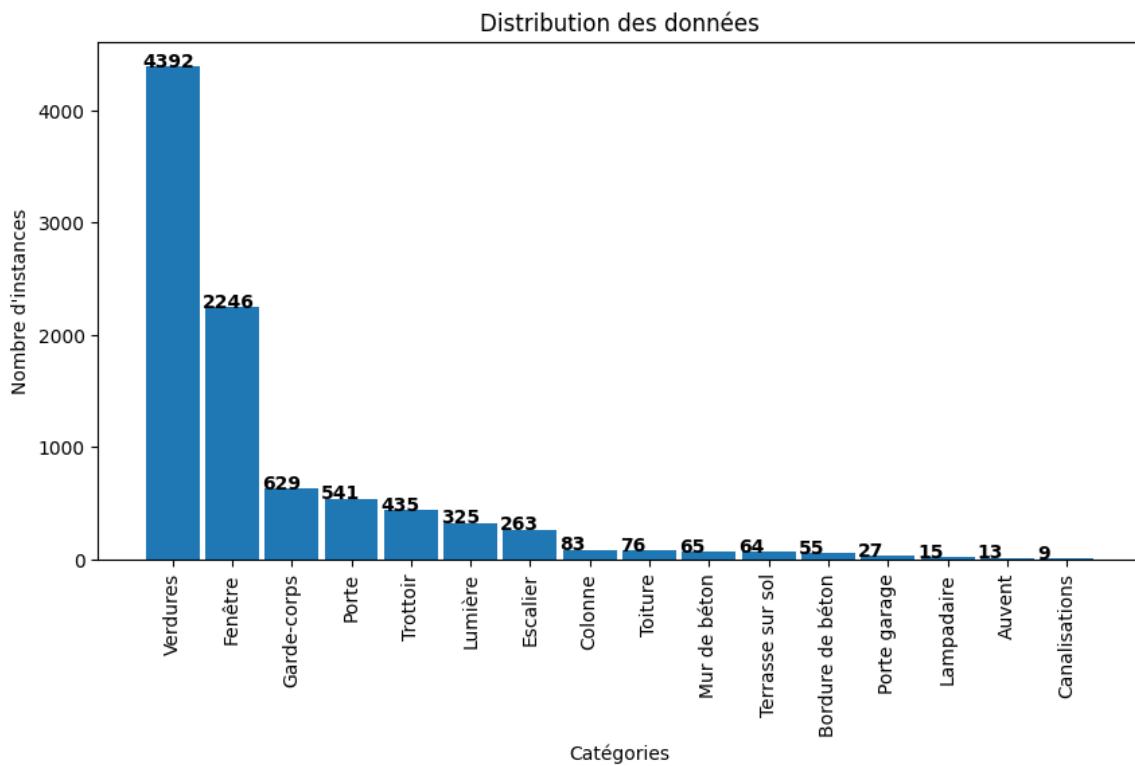


FIGURE 3.4 – Distributions des données avant la divisions en jeux d’entraînement et de validation

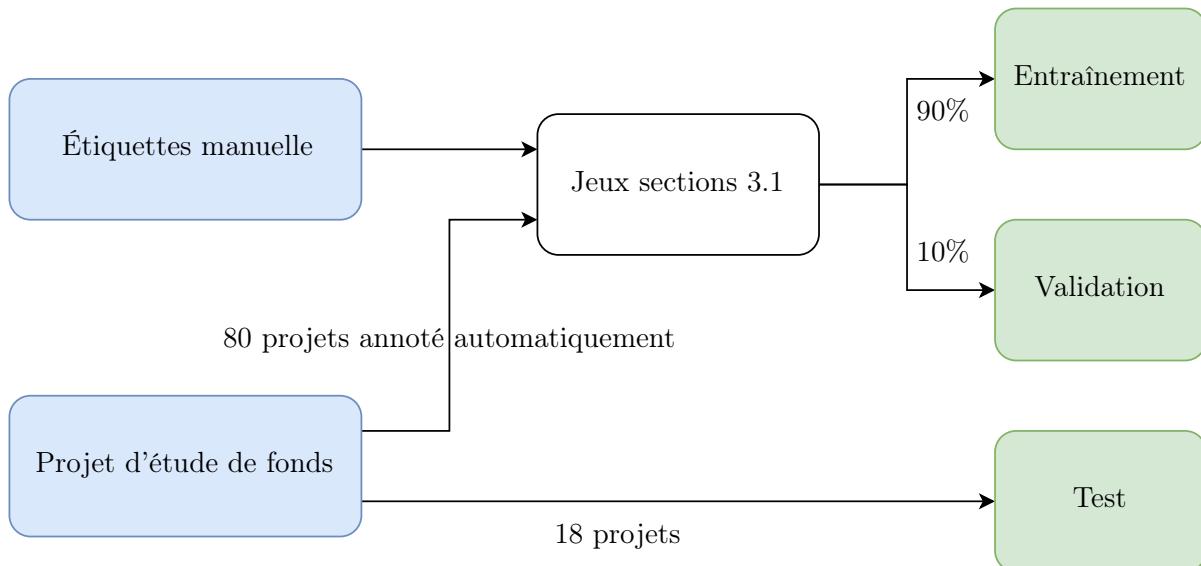


FIGURE 3.5 – Schéma décrivant la divisions des jeux de données.  
Pour plus d’information sur les distributions des jeux du  
projet (Annexe D)

Le jeu de données décrit dans la section 3.1 est comme expliqué plus haut très débalancé. En effet, dans la figure 3.4 on peut regrouper les catégories en trois grands groupes qu'on utilisera pour les discussions et l'analyse des résultats.

- 1. Le groupe des catégories fréquentes, avec les verdure et les fenêtres qui à elles deux représentent environ 75% du jeu de données.
- 2. Le groupe des catégories très rare, ce sont les catégories avec moins de 20 instances, ensemble elles ne représentent qu'environ 0.5 % du jeu de données. Elles sont présentes dans le jeu, mais on n'espère pas obtenir de résultats avec elles.
- 3. Les groupes des catégories rares, ce sont le reste des catégories, les méthodes sur le débancement étudiées auront pour but d'améliorer les résultats sur ces classes.

Enfin, pour la répartition des jeux décrite dans la figure [pipeline jeux], le jeu d'entraînement représente 90 % du jeu décrit ci-dessus et le jeu de validation représente donc les 10 % restants, la classe la plus rare du jeu (canalisations) ayant seulement 9 instances, il a été choisi de ne pas l'inclure dans le jeu de validation, c'est donc pour cela que dans la section 5 on ne la retrouvera pas dans la synthèse des résultats. Pour les données de test, cela correspond à 18 projets de Hoodi.ai avec en moyenne une trentaine d'images par projet.

# Chapitre 4

## Baselines

Dans ce projet, la problématique principale est le manque de données ainsi que le débordement de celles-ci (Chapitre 3). Afin de combattre ces obstacles, une des idées principales est le transfert d'apprentissage. Cette méthode sera réalisée avec des modèles de detectron2 (Wu et al., 2019a) ainsi que de l'augmentation de données afin d'augmenter artificiellement les données en plus de rendre les modèles plus robustes aux différentes caractéristiques que peuvent avoir les photos prises par les clients (luminosités différentes, rotation de l'appareil, etc.). Deuxièmement, le rééchantillonnage de plus que l'utilisation d'une focal loss permettra de rebalancer artificiellement le jeu et de rendre les modèles plus robustes quant à la prédiction des classes les plus rares dans le jeu.

### 4.1 Modèles

Dans ce rapport, les modèles de baseline présentés seront les modèles zoo de detectron2 suivants : segmentation d'instance Mask R-CNN R50-FPN entraînée sur COCO (R50\_COCO pour la suite), segmentation d'instance Mask R-CNN R101-FPN entraînée sur COCO (R101\_COCO) et segmentation d'instance Mask R-CNN R50-FPN entraînée sur Cityscapes (R50\_City) (Wu et al., 2019b). Plus précisément, ces différents termes signifient :

**Segmentation d'instances** La segmentation d'instances permet de détecter et de placer précisément les différentes instances du modèle dans une image. Dans le contexte de ce projet, les instances seront les différentes composantes du bâtiment à savoir les 16 catégories listées dans l'annexe C.

**Mask R-CNN** Mask R-CNN est un algorithme de détection d’objet qui est capable d’utiliser les masques des objets pour son entraînement. Cet algorithme a été choisi pour cette caractéristique, car les classes du projet ont des formes très variées et parfois complexe (verdure par exemple), cela permettra donc d’utiliser ces caractéristiques pour obtenir de meilleurs résultats.

**R50 et R101** Diminutif pour RetinaNet50 et RetinaNet101. Ces architectures sont utilisées comme base des modèles de la baseline.

**FPN (Feature Pyramidal Network)** Système pyramidal standard à base de couches de convolution et de réseaux de neurones complètement connecté. Ces couches sont utilisées par-dessus les architectures R50 et R101.

## 4.2 Méthode

Ensuite, les différentes méthodes utilisées afin de répondre à la problématique sont l’augmentation de données, l’échantillonnage et la focal loss.

### 4.2.1 Augmentation de données

Le jeu de données étant très faible en quantité d’instances par classe, des fonctions d’augmentation de données seront utilisées. La liste est la suivante :

- • RandomBrightness
- • RandomContrast
- • RandomSaturation
- • RandomRotation

Chacune de ces fonctions permet à la fois d’augmenter artificiellement les données, mais aussi de modéliser le côté aléatoire des photos que le client peut fournir. Les trois premières fonctions permettent de modifier la luminosité, le contraste et la saturation des photos qui peuvent être différents en fonction de l’heure ainsi que de la saison de la prise de la photo. Pour la fonction qui modifie aléatoirement la rotation de l’image, cela permet de modéliser l’angle dans lequel le client tient son appareil lors de la capture de sa photo.

Les hyperparamètres de cette méthode seront donc le taux de luminosité, de contraste, de saturation minimal et maximal ainsi de l’angle de rotation minimal et maximal de l’image.

#### 4.2.2 Échantillonnage

Ensuite, pour le rebalancement des données, une des méthodes utilisées dans ce rapport est le rééchantillonnage des données. Dans ce projet, on rajoutera naïvement une seconde fois les instances lorsque la catégorie représente moins d'un certain seuil du jeu de données (Figure 4.1).

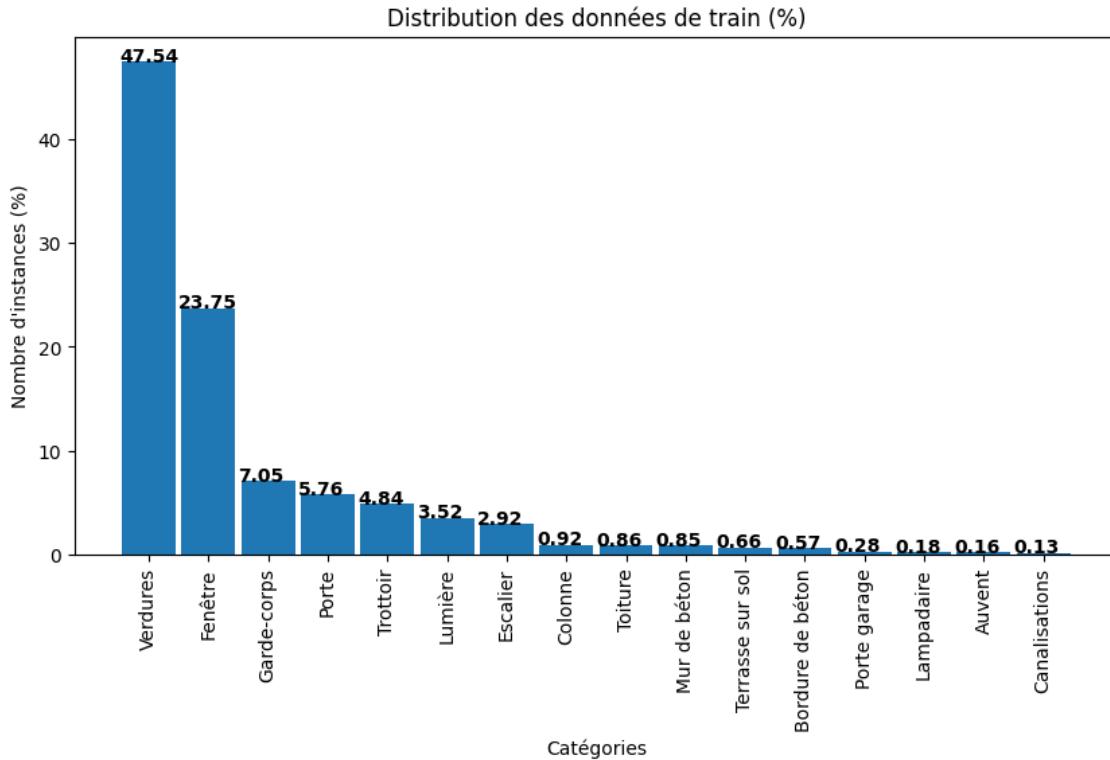


FIGURE 4.1 – Distributions des données du jeu d’entraînement en pourcentage

L’hyperparamètre de cette méthode est fixé à un seuil de 10 % afin d’y ajouter à nouveau les annotations de toutes les classes à l’exception des deux classes les plus fréquentes.

#### 4.2.3 Ajustement de la fonction de loss

Enfin, la seconde méthode pour le rebalancement des données est similaire à la méthode de rééchantillonnage. Cependant, au lieu d’ajouter des données, on pénalisera plus ou moins fortement les modèles lors de l’entraînement en fonction de la rareté des catégories. Les hyperparamètres de cette méthode sont donc  $\alpha$  et  $\gamma$ .

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t)$$

$p_t$  : pourcentage de confiance de la prédiction du modèle

## 4.3 Métriques

Les principales métriques utilisées pour comparer nos modèles sont différentes lors de la phase de validation et la phase de test des modèles.

Pour la phase de validation des modèles, les métriques utilisées sont la précision moyenne (AP) ainsi que l'intersection entre l'aire de prédiction et celle réelle sur l'union des deux (IoU). Cependant, dans le contexte de ce projet, on cherche à établir une liste des composantes présentes sur la façade des bâtiments. Leurs positions exactes ainsi que leurs nombres ne nous intéressent donc pas, donc nous préférerons comparer nos résultats avec un IoU supérieur à un ratio de 0.5. Lors de la présentation des résultats, on verra donc apparaître plusieurs termes : (AP, AP50, mAP). Pour plus de détails dans la présentation des résultats, on comparera ces métriques sur la boîte englobante des instances (bbox) et sur leurs masques (mask).

**AP** Précision moyenne d'une classe

**AP50** Précision moyenne d'une classe pour les prédictions avec un IoU supérieur à un ratio de 0.5

**mAP** La moyenne des AP de chaque classe. Chacune des classes ont le même poids dans la pondération de la moyenne.

Pour la phase de test, la métrique utilisée sera principalement la précision. Dans le cadre de ce projet, on préférera avoir une bonne précision plutôt qu'un rappel, car dans le contexte d'implémentation sur le site web de l'entreprise, le fait de se tromper dans la prédiction d'une classe impliquerait une action du client afin de supprimer la composante erronée. De plus, une étude de fonds de prévoyance comporte en général 20 à 30 photos de la copropriété l'objectif de cette implémentation est de demander au maximum au client une suppression d'une seule composante. Cela revient donc à une précision d'environ 96 % sur la prédiction de l'inventaire des composantes.

## 4.4 Résultats de la baseline

Enfin, dans les résultats des méthodes décrites ci-dessus, seront comparés aux résultats décrits dans le tableau 4.1. Ces résultats représentent la base de comparaison des métriques (Sous-section 4.3) pour les méthodes (Sous-section 4.2) sur chacun des modèles de la baseline (Sous-section 4.1).

Baseline	mAP bbox	mAP50 bbox	mAP masque	mAP50 masque
R50_COCO	14.02	22.31	14.51	20.96
R101_COCO	14.35	22.29	14.35	19.38
R50_CITY	7.37	11.81	7.15	10.59

TABLE 4.1 – **Segmentation d’instance** : mAP des classes de l’étude sur les différentes baselines de l’étude

En comparaison du modèle pré-entraîné sur Cityscapes (Cordts et al., 2016) les modèles pré-entraînés sur le jeu de donnée COCO (Lin et al., 2014) obtiennent en moyenne une meilleure précision. Cependant, les backbones des modèles semblent pour l’instant ne pas être déterminante sur la précision des modèles.

	Auvent	Lampadaire	Porte garage	Bordure de béton	Terrasse sur sol
R50_COCO	0.0	0.0	54.65	5.77	13.47
R101_COCO	0.0	0.0	36.01	10.98	19.36
R50_CITY	0.0	0.0	21.87	3.00	0.34

	Mur de béton	Toiture	Colonne	Escalier	Lumière
R50_COCO	0.00	6.03	3.37	14.86	16.81
R101_COCO	1.06	4.95	8.81	16.05	19.50
R50_CITY	0.00	0.59	2.11	9.74	11.05

	Trottoir	Porte	Garde-corps	Fenêtre	Vertures
R50_COCO	11.83	17.00	8.40	43.16	25.99
R101_COCO	13.83	20.59	6.60	44.25	27.53
R50_CITY	5.62	6.14	3.06	25.34	24.43

TABLE 4.2 – **Segmentation d’instances** : AP pour chacune des classes de l’étude sur chaque baseline

Les classes sont arrangées de la moins présente à la plus présente dans le jeu de l’étude. Figure 3.4

La classe "Canalisation" n'est pas présente dans le tableau, car dû à son nombre d'itération trop faible elle n'est pas dans le jeu de validation.

Pour les détails des classes, on remarque ici qu'une classe peu présente dans le jeu ressort tout de même avec de bons résultats cela peut s'expliquer dû à sa forme et sa taille. En général dans le cadre d'une étude de fonds de prévoyance, les clients prennent seulement cet élément précis en photos. Les portes de garages seront donc souvent en plein milieu de l'image et en grand. De plus, comme on peut s'y attendre, les catégories à nombre d'instances très faible ne sont absolument pas détectées et ont une précision moyenne de 0 %.

# Chapitre 5

## Expérimentation

### 5.1 Synthèse des résultats

		mAP bbox	mAP50 bbox	mAP masque	mAP50 masque
R50_COCO	Échantillonnage	14.83	22.41	15.43	22.58
	Augmentation	7.40	13.84	8.49	12.98
	Focal loss	14.26	22.33	14.45	21.78
R101_COCO	Échantillonnage	15.21	22.44	15.31	21.78
	Augmentation	9.39	16.58	11.00	16.32
	Focal loss	15.00	22.81	15.25	22.37
R50_CITY	Échantillonnage	6.78	10.60	7.03	10.45
	Augmentation	4.66	9.43	6.12	9.37
	Focal loss	7.46	12.20	8.27	12.22

TABLE 5.1 – Synthèse des résultats en mAP des classes de l'étude sur les différentes méthodes de l'étude. Les meilleurs résultats de chaque backbone sont indiqués en gras.

Pour R50\_COCO : Méthode de l'échantillonnage avec un gain de 1.62% de précision moyenne avec un IoU  $\in [0.5, 1]$  sur les masques.

Pour R101\_COCO : Méthode du focal loss avec un gain de 2.99% de précision moyenne avec un IoU  $\in [0.5, 1]$  sur les masques.

Pour R50\_CITY : Méthode du focal loss avec un gain de 1.63% de précision moyenne avec un IoU  $\in [0.5, 1]$  sur les masques.

		Auvent	Lampadaire	Porte garage	Bordure de béton	Terrasse sur sol
R50_COCO	Échantillonnage	0.0	0.0	54.98	13.60	15.15
	Augmentation	0.0	0.0	23.17	0.00	11.64
	Focal loss	0.0	0.0	55.74	6.82	14.51
R101_COCO	Échantillonnage	0.0	0.0	37.62	20.56	18.85
	Augmentation	0.0	0.0	23.17	0.00	11.64
	Focal loss	0.0	0.0	41.03	12.35	21.78
R50_CITY	Échantillonnage	0.0	0.0	27.75	1.80	1.12
	Augmentation	0.0	0.0	0.00	0.00	0.00
	Focal loss	0.0	0.0	21.88	5.92	2.89
		Mur de béton	Toiture	Colonne	Escalier	Lumière
R50_COCO	Échantillonnage	0.2	11.73	5.94	15.77	14.59
	Augmentation	0.0	0.0	0.28	2.49	14.43
	Focal loss	0.0	8.41	6.90	14.99	18.34
R101_COCO	Échantillonnage	2.07	3.59	8.15	18.28	18.69
	Augmentation	0.0	0.0	0.28	4.81	16.87
	Focal loss	0.0	6.54	9.41	16.63	15.47
R50_CITY	Échantillonnage	0.00	0.00	0.61	6.99	8.39
	Augmentation	0.0	0.0	0.00	1.59	9.79
	Focal loss	0.0	0.00	3.02	11.45	9.99
		Trottoir	Porte	Garde-corps	Fenêtre	Verdures
R50_COCO	Échantillonnage	11.52	18.17	4.91	43.33	27.24
	Augmentation	4.72	17.38	4.34	50.64	26.82
	Focal loss	10.49	17.16	5.66	43.89	25.29
R101_COCO	Échantillonnage	15.22	19.51	6.96	43.75	28.81
	Augmentation	5.67	19.99	8.11	51.58	26.66
	Focal loss	14.34	20.05	7.81	46.29	28.45
R50_CITY	Échantillonnage	5.30	5.66	2.92	23.70	23.12
	Augmentation	3.06	15.61	2.18	36.21	23.36
	Focal loss	5.30	7.49	3.27	24.58	23.62

TABLE 5.2 – Résultats en AP (en %) de chacune des classes de l'étude pour chacune des méthodes d'**échantillonnage**, d'**augmentation des données** et de **focal loss**.

En vert, on retrouve les gains de précisions par rapport à la baseline.

En rouge, les pertes de précisions par rapport à la baseline.

## 5.2 Analyse et discussion des résultats

De manière générale, les méthodes pour contrer le débalancement de données apportent un gain en précision aux modèles. Cependant, ce gain reste faible (0.8 % sur les boîtes englobantes pour la méthode d'échantillonnage et 0.7 % sur les boîtes englobantes pour la méthode d'ajustement de la fonction de loss) et cela peut s'expliquer par le nombre d'instances présentes dans les classes. De plus, dépendamment de la taille de l'objet, et de sa position dans les images d'entraînement et de validation, la précision peut fortement varier. C'est le cas pour les portes de garage, la précision est haute. Ceci peut se voir dans le tableau 5.2 dans les expériences d'augmentation de données. La précision de la classe baisse fortement, car l'augmentation de données va changer l'emplacement de l'objet dans l'image. Plus précisément, pour chacune des méthodes de la sous-section 4.2.

### 5.2.1 Échantillonnage

Pour les modèles pré-entraînés sur le jeu de donnée COCO, on observe un gain en mAP d'environ 0.8 %, pour les masques ce gain est même supérieur (environ 1 % pour les boîtes et 2 % pour les masques). Cela peut s'expliquer par le tableau 5.2 dans la partie échantillonnage. En effet, on observe des gains significatifs en AP et AP50 pour les boîtes englobantes et les masques dans les classes les plus rares. On observe aussi trois regroupements de classes : les très rares pour lesquels la précision moyenne reste à 0 % les moins rares pour lesquels la précision moyenne augmente et les fréquentes pour lesquels la précision moyenne diminue ou augmente peu. Contrairement au transfert d'apprentissage sur COCO, le transfert d'apprentissage sur Cityscapes obtient de moins bons résultats avec cette méthode, on pourrait expliquer cela par l'architecture des bâtiments et de ses caractéristiques, mais cela n'est qu'une supposition.

### 5.2.2 Focal loss

Pour les modèles pré-entraînés sur le jeu de donnée COCO, on observe un gain en mAP d'environ 0.2 %, pour les masques ce gain est même supérieur (1 % pour les masques). Cela peut s'expliquer par le tableau 5.2 dans la partie focal loss. Les résultats et l'observation sont tout deux très similaires à la méthode précédente.

### 5.2.3 Augmentation des données

Enfin, pour l'augmentation des données, on observe aucun gain en précision moyenne. Au contraire, on remarque même une perte de précision. La plus flagrante étant celle sur la classe « porte de garage » pour la raison expliquée dans l'introduction de ce chapitre. De plus, dans les trois types de classes identifiées (très rare, rare et fréquente) on observe aucun changement pour les très rares, des baisses de précision moyenne dans les classes rares mais des gains dans les classes fréquentes. Cela est donc dérangeant pour la résolution du problème de débalancement.

## 5.3 Recherche d'hyperparamètre

Pour la recherche d'hyperparamètres, cela a été effectué par tâtonnement sur le modèle R50\_COCO. Les hyperparamètres sélectionnés sont donc les hyperparamètres pour lesquels le modèle a obtenu le meilleur AP sur le jeu de validation. Les hyperparamètres par méthodes sont listés dans le tableau 5.3.



### 5.4.3 R50\_CITY

Augmentation de données			Échantillonnage			Focal loss		
	En réalité			En réalité			En réalité	
	Objet	Pas d'objet		Objet	Pas d'objet		Objet	Pas d'objet
Prédictions	Objet	88	16	Prédictions	Objet	113	26	Prédictions
	Pas d'objet	89	95		Pas d'objet	64	85	Objet
Accuracy	63.54%		Accuracy	68.75%		Accuracy	68.06%	
Précisions	84.62%		Précisions	81.29%		Précisions	81.48%	
Rappel	49.72%		Rappel	63.84%		Rappel	62.15%	
F1-Score	62.63%		F1-Score	71.52%		F1-Score	70.51%	

TABLE 5.6 – Matrices de confusion générale pour le modèle R50\_CITY

### 5.4.4 Discussions

Les tableaux 5.4, 5.5 et 5.6 indiquent clairement qu'en termes de précisions les meilleurs résultats sont obtenus par le modèle R101\_COCO. Cependant, si on regarde la matrice de confusions, on remarque que sur toutes les instances des 18 projets de test le modèle n'a que réussi à prédire 29 objets correctement pour la méthode d'échantillonnage. En regardant le rappel on remarque donc que maximiser la précision a rendu le modèle très craintif pour prédire des instances. Il faut donc trouver un moyen de prendre en compte le rappel pour la sélection du meilleur modèle.

### 5.4.5 F $\beta$ -Score

Il existe le F1-Score qui est très connu pour calculer un score en prenant compte de la précision et du rappel. Cependant, ce score accorde la même importance à la précision et le rappel, or ce n'est pas intéressant dans le contexte de ce projet. On veut tout de même garder une importance plus forte à la précision. On calculera donc le F $\beta$ -Score avec un  $\beta$  inférieur à 1 pour donner plus d'importance à la précision.

$$F\beta = \frac{(1+\beta^2)PR}{\beta^2P+R}, \quad \beta = 0.5, P : \text{Précisions}, R : \text{Rappel}$$

Modèles	Méthodes	F $\beta$ -Score
<b>R50_COCO</b>	Augmentation	77.14%
	Échantillonnage	77.25%
	<b>Focal loss</b>	<b>77.78%</b>
R101_COCO	Augmentation	68.51%
	Échantillonnage	46.32%
	Focal loss	49.85%
R50_CITY	Augmentation	73.72%
	Échantillonnage	77.07%
	Focal loss	76.71%

TABLE 5.7 – F0.5-Scores des modèles pour chaque méthodes De cette manière, on obtient donc le meilleurs modèle R50\_COCO avec la méthode d'ajustement de la fonction de loss

#### 5.4.6 Inférences

En effectuant l'inférence avec R50\_COCO, on obtient les figures 5.1. Cependant, on observe sur certains résultats des limites. Le modèle est bon dans la prédiction d'instances pour des images ciblées sur des endroits particuliers comme des balcons ou rampes d'escaliers (Figures 5.1a, 5.1b et 5.1c). Mais lorsque la photo est plus générale et représente une façade d'un bâtiment alors le modèle aura plus de mal à détecter toutes les composantes (Figures 5.1d et 5.1e). En tentant d'entraîner des modèles afin de maximiser la précision, les modèles sont devenus très craintifs pour ce qui est de prédire et ne préfèrent ne rien prédire plutôt que de risquer de se tromper. Il faut donc prendre en compte le rappel pour la sélection du meilleur modèle.

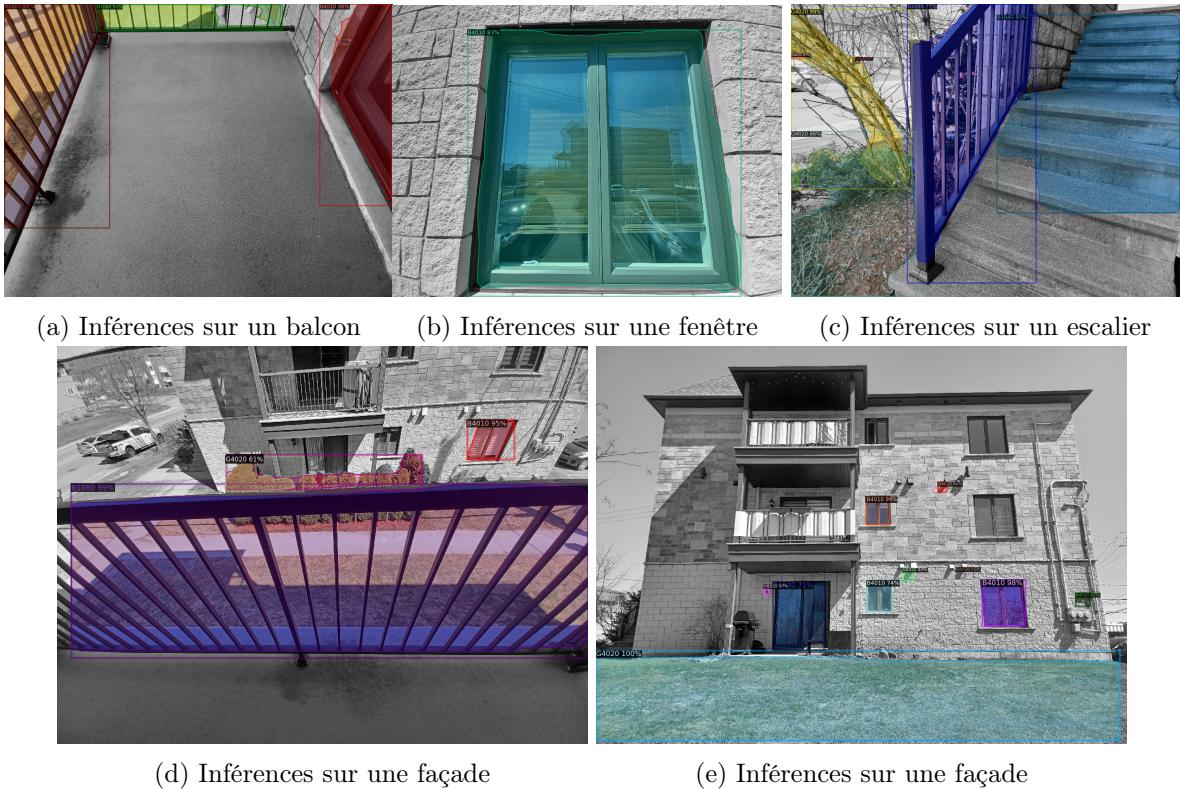


FIGURE 5.1 – Exemples d'inférences avec le modèles R50\_COCO et la méthode d'ajustement de la fonction de loss

## 5.5 Ouverture

Enfin, dans le cadre du stage, l'objectif des 96 % de précision sur le test n'a pas été atteint, car on arrive aux alentours des 80 % de précisions. De plus, il faut prendre en compte que le modèle prédit aussi des instances hors propriété (Figure 3.3c) et cela n'est absolument pas envisageable dans le cadre d'une mise en production. En outre, dans un contexte de mise en production, le client postera toutes ses photos sur le site de manière à ce qu'elles soient traitées (soit une vingtaine/trentaine d'images). Mais le mask R-CNN est lent pour l'inférence des images d'un projet complet cela est très variable en fonction de son backbone ou de la taille des images en entrée, mais il faut compter environ 1 minute et 30 secondes par projet pour le R50 et 2 minutes et 30 secondes pour R101 avec la puissance de calcul du site de l'entreprise.

Pour finir, ce qui a été proposé à l'entreprise pour continuer le projet n'est en priorité pas de collecter plus de données, car il y a déjà plus de 2000 images, mais d'abord de corriger les annotations (oublis, imperfections). De plus, une autre piste pour ce projet est de se pencher sur la gestion des instances hors propriété. Des pistes de solutions ont été proposées comme détecter les limites de la façade du bâtiment avant de commencer la détection de composantes ou sinon de calculer la profondeur des composantes à l'aide de l'image et des plans du bâtiment du carnet d'entretien, cependant certaines copropriétés étant vieilles leurs plans peuvent ne pas être soumis aux mêmes normes. Par la suite, lorsque la catégorie sera prédite, il faudra la classifier dans sa bonne classe, car les classes ont été regroupées. Cela peut être fait par du post-traitement en utilisant les métadonnées prédites par le modèle comme l'aire de l'objet ainsi que son matériau, pour le moment encore inutilisés.

AP	AP50	AP75	APs	APm	APl
67.67	80.23	75.71	2.48	60.41	78.13

TABLE 5.8 – mAP de la classe fenêtre en utilisant R50\_COCO

AP	AP50	AP75	APs	APm	APl
20.94	26.66	24.19	0.00	10.05	23.43

TABLE 5.9 – mAP de fenêtres, garde-corps et portes en utilisant R50\_COCO

Fenêtre	Garde-corps	Portes
62.81	0.00	0.00

TABLE 5.10 – AP par classes pour les 3 classes

Une dernière idée proposée à l'entreprise est de revoir entièrement la résolution du problème. Au lieu d'entraîner un seul modèle qui prédit toutes les classes, l'algorithme pourrait être plusieurs modèles qui correspondent à chaque classe. En effet, une expérience a été faite durant ce stage pour prédire seulement une classe avec le modèle (Tableau 5.8).

Les résultats sont bien meilleurs de cette manière. De plus, pour tester les limites, plusieurs classes ont été ajoutées pour voir l'évolution du mAP lorsqu'on rajoute deux classes (Tableaux 5.9 et 5.10). Cependant, si cette structure est envisagée, il faudra revoir les modèles à utiliser, car plus on ajoutera de classes plus on ajoutera de modèles et avec des algorithmes de détection mask R-CNN l'inférence deviendra trop longue dans un contexte de mise en production. Les algorithmes YOLO seraient donc une bonne alternative pour cette structure, mais dans ce cas la problématique du manque d'annotations reviens sur la table, car une centaine d'instances est vraiment trop peu même avec du transfert d'apprentissage d'après la documentation de YOLO (ultralytics, 2022).

# Chapitre 6

## Conclusion

En conclusion, la problématique qui était de détecter et de classifier les composantes d'une façade d'un bâtiment afin d'établir un inventaire du bâtiment dans un contexte d'études de fonds de prévoyance n'a été que partiellement répondu. En effet, les résultats du chapitre 5.2 montrent que l'on est capable d'établir une liste de composantes proche de la réalité, mais ce n'est pas encore suffisant pour le mettre en place sur le site de l'entreprise (précision de 80 % au lieu de 96 %). Il pourrait cependant être utilisé afin de proposer les composantes pendant la phase de sélection manuelle au lieu de la remplacer. En ce qui concerne le futur du projet, une phase d'annotations plus propre est une priorité majeure pour pouvoir obtenir de meilleurs résultats, car les annotations rares ne sont pas détectées et font baisser la précision des modèles. Finalement, l'architecture de l'algorithme peut aussi être remise en question. Car Mask R-CNN lorsqu'il y a beaucoup de classes et peu de données, il a tendance à regrouper les classes par formes. Par exemple, il confondra les portes et la fenêtre, mais pas les verdures, car elles ont des formes atypiques. Un modèle par classe ou part groupement faible de classe similaire serait une bonne piste de progression.

# Bibliographie

- I. O. Agyemang, X. Zhang, I. A. Mensah, B. C. Mawuli, B. L. Y. Agbley, and J. R. Arhin. Enhanced deep convolutional neural network for building component detection towards structural health monitoring. In *2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pages 202–206, 2021. doi : 10.1109/PRAI53619.2021.9551102.
- P. Bharati and A. Pramanik. Deep learning techniques—r-cnn to mask r-cnn : A survey. 2020. doi : 10.1007/978-981-13-9042-5\_56.
- M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- H. Face, 2022. URL [https://huggingface.co/models?pipeline\\_tag=object-detection](https://huggingface.co/models?pipeline_tag=object-detection).
- U. Handalage and L. Kuganandamurthy. Real-time object detection using yolo : A review. 2021. doi : 10.13140/RG.2.2.24367.66723.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. URL <https://arxiv.org/abs/1512.03385>.
- K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL <http://arxiv.org/abs/1703.06870>.
- J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi. OneFormer : One Transformer to Rule Universal Image Segmentation. 2023.
- J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 2019. URL <https://doi.org/10.1186/s40537-019-0192-5>.
- T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO : common objects in context. *CoRR*, abs/1405.0312, 2014. URL <http://arxiv.org/abs/1405.0312>.

- C. Liu, S. Sepasgozar, S. Shirowzhan, and G. Mohammadi. Applications of object detection in modular construction based on a comparative evaluation of deep learning algorithms. *Construction Innovation*, ahead-of-print, 05 2021. doi : 10.1108/CI-02-2020-0017.
- S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN : towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. URL <http://arxiv.org/abs/1506.01497>.
- ultralytics. Tips for best training results, 2022. URL <https://github.com/ultralytics/yolov5/wiki/Tips-for-Best-Training-Results>.
- Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2 - data augmentation. <https://github.com/facebookresearch/detectron2>, 2019a.
- Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. [https://github.com/facebookresearch/detectron2/blob/main/MODEL\\_ZOO.md](https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md), 2019b.
- B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20K dataset. *CoRR*, abs/1608.05442, 2016. URL <http://arxiv.org/abs/1608.05442>.

## Annexe A

# Architecture

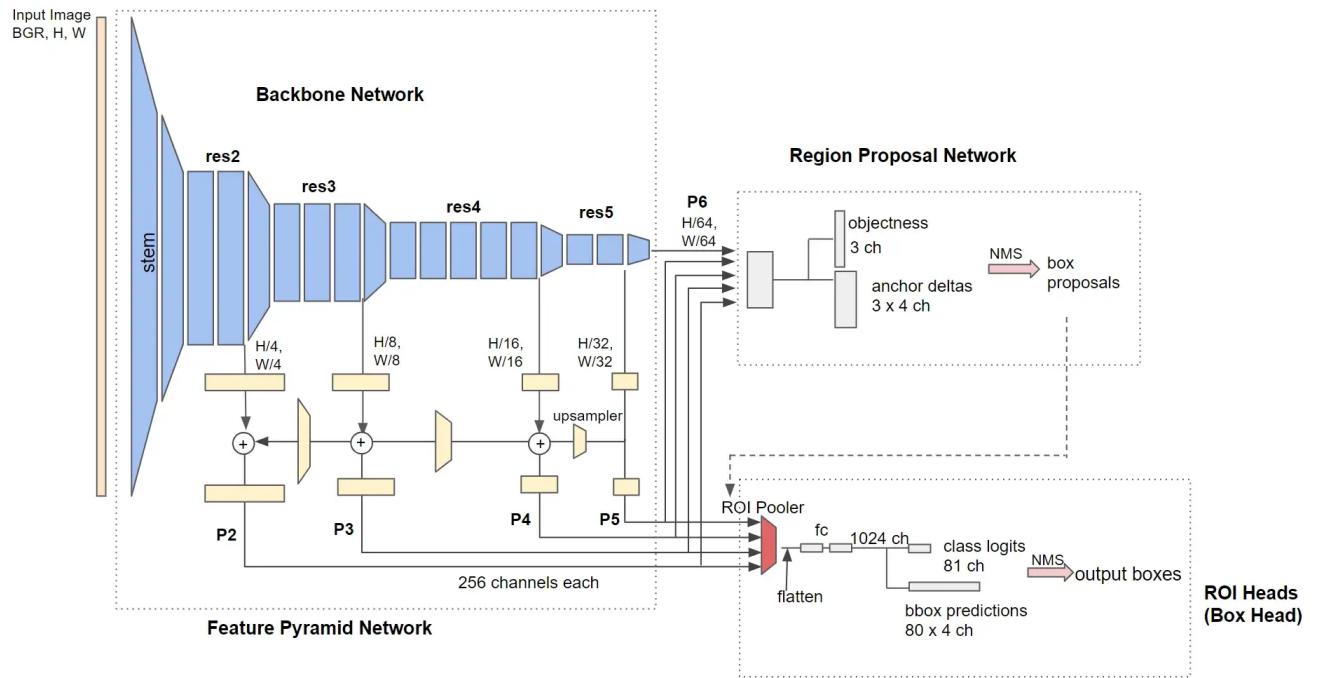


FIGURE A.1 – Architecture du modèle Mask R-CNN : R50-FPN de Detectron2

Source : <https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd>

## Annexe B

### Liste des classes hiérarchisées

Composantes (Niveau 1)	Composantes avec matériaux (Niveau2)
Murs de béton coulé en place	Murs de béton coulé en place
Dalles des balcons	Dalle de béton structurale des balcons Dalles des balcons en béton préfabriquées
Platelage/Pontage des balcons	Platelage des balcons en planches de bois Pontage de balcon en fibrociment Platelage des balcons en planches aluminium Pontage de balcon, coursive et marches extérieures en fibre de verre
Structure de support	Structure en aluminium des escaliers et paliers extérieurs d'issues Structure en acier galvanisé des escaliers et paliers extérieurs d'issues Structure des balcons en acier galvanisé Colonnes de support des balcons en bois avec recouvrement métallique Colonnes de support des balcons en aluminium Structures de support en bois des balcons Structure des balcons en acier peint Structure en acier peint des escaliers et paliers extérieurs d'issues Colonnes en acier peint
Escaliers, paliers	Escaliers, paliers et rampes d'accès extérieurs en béton

TABLE B.1 – Listes des classes partie 1

Niveau 1 : 44 classes

Niveau 2 : 81 classes

<b>Composantes (Niveau 1)</b>	<b>Composantes avec matériaux (Niveau2)</b>
Terrasses sur sol	Terrasses sur sol en pavés imbriqués
	Terrasses sur sol en béton
	Terrasses sur sol en bois
	Terrasses en bois sur toit
	Terrasse sur toit en dalles modulaires sur plots
Garde-corps	Garde-corps en bois
	Garde-corps en plastique (PVC)
	Ecrans séparateurs de balcons
	Garde-corps en verre et acier/métal
	Garde-corps en aluminium
	Garde-corps en acier peint
Revêtement	Revêtement de maçonnerie à cavité
	Système d'isolation des façades avec enduit (SIFE)
	Revêtement à clins de bois
	Revêtement de fibrociment
	Revêtement à clins d'aluminium
	Revêtement extérieur en panneaux métalliques
	Revêtements à clins de vinyle
	Revêtement de bois d'ingénierie de type CanExel
Murs rideaux	Murs rideaux
Toiture	Toiture métallique
	Toiture inclinée en bardeau d'asphalte
	Toiture à membrane élastomère
Gouttières et descentes pluviales	Gouttières et descentes pluviales
Toiture inversée	Toiture inversée
Fenêtres	Fenêtres en aluminium
	Fenêtres en PVC
Portes-fenêtres coulissantes	Portes-fenêtres coulissantes en aluminium
	Portes-fenêtres coulissantes en PVC
Portes-terrasses	Portes-terrasses avec cadre en PVC ou aluminium
Porte d'entrée	Porte d'entrée extérieure en bois
	Porte à profilés d'aluminium
	Porte d'entrée principale commune en acier
	Portes extérieures en acier
Portes de garage	Porte élévatrice sectionnelle (portes de garage)
Produit de calfeutrage des ouvertures	Produit de calfeutrage des ouvertures
Mobiliers extérieurs	Mobiliers extérieurs
	Casier postal à usage extérieur
	Bacs de plantations en aluminium

TABLE B.2 – Listes des classes partie 2

Niveau 1 : 44 classes

Niveau 2 : 81 classes

<b>Composantes (Niveau 1)</b>	<b>Composantes avec matériaux (Niveau2)</b>
Bassin de piscine	Bassin de piscine en béton avec finition en carrelage
	Bassin de piscine en béton avec fini de peinture
	Bassin de piscine avec toile de vinyle
Plage de béton	Plage de béton des aménagements aquatiques
Équipements de filtration	Équipements de filtration de l'eau de la piscine
Auvent de toile	Auvent de toile sur une structure métallique
Remises	Remises extérieures préfabriquées
	Remise à ossature de bois
Chaussée en enrobé bitumineux	Chaussée en enrobé bitumineux
Bordures et allées piétonnes	Bordures de béton
	Allées piétonnes en béton
	Allées piétonnes en pavés imbriqués
Escaliers en bois	Escaliers en bois
Clôture	Clôture à mailles de chaîne et lattes d'intimité
	Clôture de bois traité
Murs de soutènement	Murs de soutènement en pierre
	Murs de soutènement en bois
	Murs de soutènement en blocs de béton
	Murs de soutènement en béton
Arbres, plantes et couvre-sol	Arbres, plantes et couvre-sol
Canalisations souterraines	Canalisations souterraines, puisards et regards de nettoyage
Caniveau de drainage en béton	Caniveau de drainage en béton
Distribution électrique	Distribution électrique aux espaces extérieurs de stationnement
Éclairage extérieur (fixé au bâtiment)	Éclairage extérieur (fixé au bâtiment)
Lampadaires de terrain	Lampadaires de terrain

TABLE B.3 – Listes des classes partie 3

Niveau 1 : 44 classes

Niveau 2 : 81 classes



## Annexe C

### Correspondances des classes en fonction de leurs code uniformat II

A1010	Murs de béton coulé en place
B1030	Balcon
B1040	Escaliers
B1050	Terrasses sur sol
B1080	Garde-corps
B3010	Toitures
B4010	Fenêtres
B4040	Portes
B4060	Portes de garage
F3010	Auvent
G1020	Bordures de béton
G2010	Trottoirs
G4020	Arbres, plantes et couvre-sol
G5020	Canalisations
G6020	Éclairage extérieur (fixé au bâtiment)
G6090	Lampadaires de terrain

TABLE C.1 – Annexe C : Descriptions des classes selon leurs annotations

# Annexe D

## Distributions

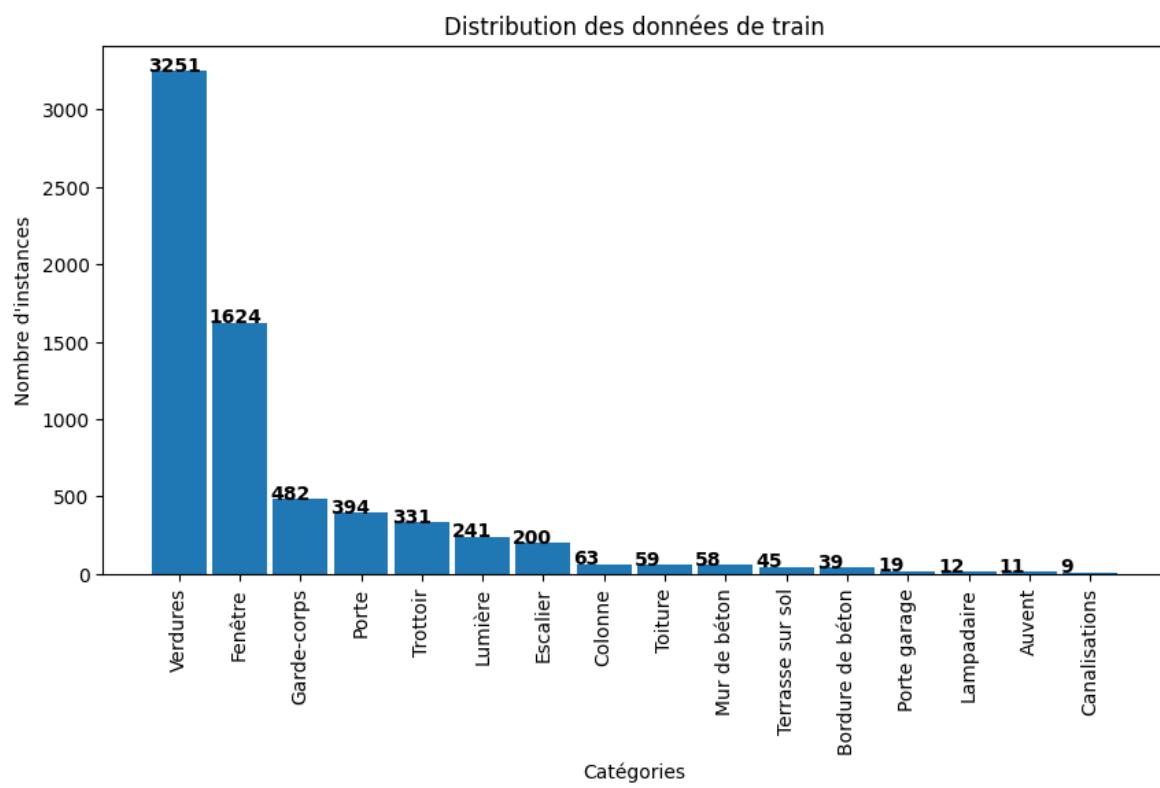


FIGURE D.1 – Distribution des données nettoyées du jeu d'entraînement

Distribution des données de val

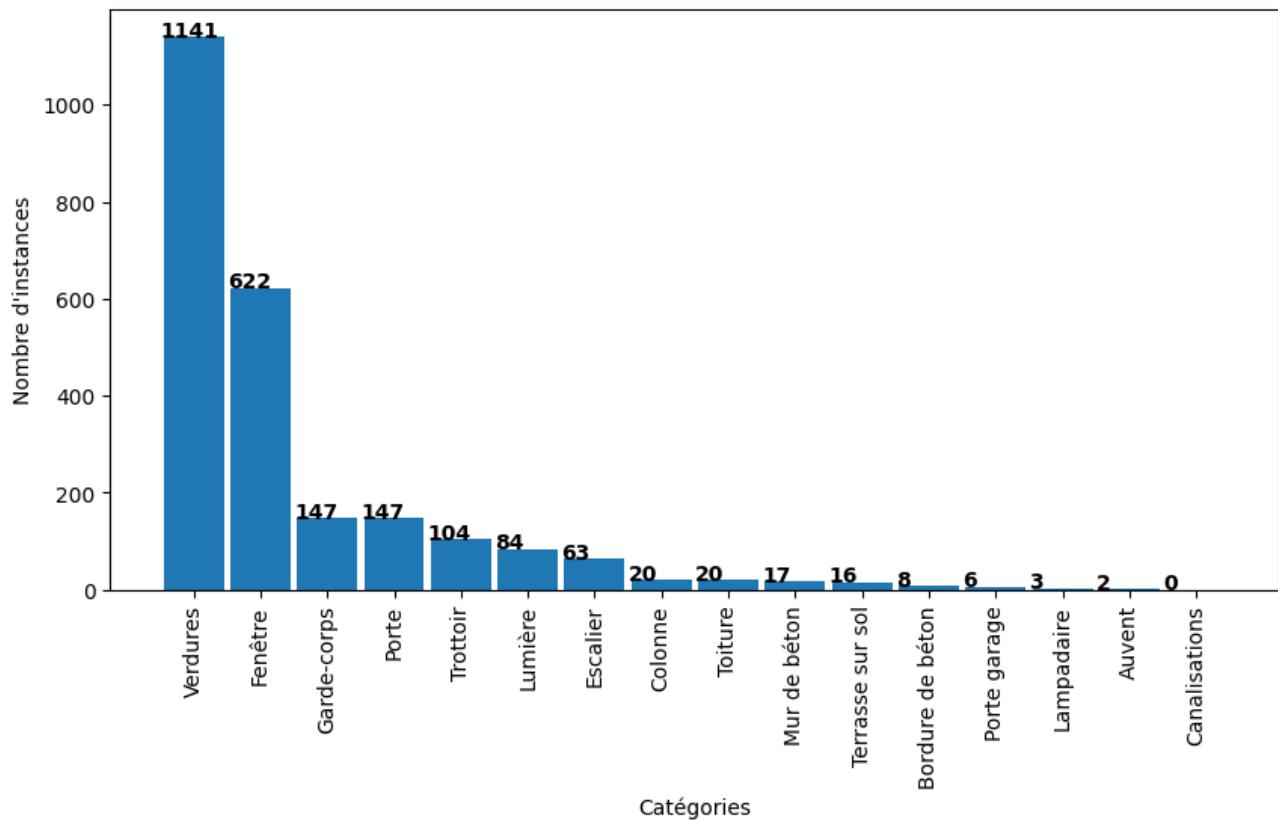


FIGURE D.2 – Distribution des données nettoyées du jeu de validation