TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ETS	École de Technologie Supérieure
LCLCL	Lettre-Chiffre-Lettre-Chiffre-Lettre
AE	Auto-Encodeur
BGR	Blue-Green-Red (Bleu-Vert-Rouge)
HSV	Hue-Saturation-Value (Teinte-Saturation-Luminosité(Valeur))
GRU	Gated recurrent unit (Réseau récurrent à portes)
CTC	Connectionist temporal classification (Classification Temporelle Connectioniste)
DAE	Deep Auto Encoder (Auto encodeur profond)

INTRODUCTION

La maintenance et le remplacement des équipements pour le transport et la distribution d'électricité sont essentiels au bon fonctionnement et à la pérennité du réseau électrique. Des évènements récents, telle que la panne majeure de novembre 2019 (figure 0.1), exposent la fragilité du réseau vieillissant. Étant donné la vaste étendue du réseau actuel et la croissance continue de la demande résidentielle, ces activités exigent d'exploiter une quantité astronomique d'information concernant la localisation et l'état des actifs. Cependant, peu d'outils sont présentement disponibles pour traiter cette information de manière automatique. Les activités de maintenance et remplacement du réseau électrique sont par ailleurs entravées par deux obstacles. En premier lieu, bien qu'Hydro-Québec possède une base de données géo-référencées des équipements de distribution (e.g., poteaux et transformateurs), l'information dans cette base de données est parfois désuète ou même erronée. Cela entraîne une perte importante de temps lors de la maintenance et le remplacement de ces équipements. De plus, l'état des équipements est présentement déterminé à l'aide d'inspections visuelles de spécialistes, un processus pouvant être long et coûteux considérant l'immensité du réseau. Afin de réduire la durée et le coût de ces activités, il est donc impératif de développer des outils pouvant fonctionner sur appareils portables pour la mise-à-jour automatisée de la localisation et l'état des équipements du réseau de distribution. L'amélioration de l'autonomie des appareils portables ainsi que le développement récent d'algorithmes puissants pour l'interprétation d'images offrent une opportunité unique de résoudre ces défis complexes de recherche.

La détection d'objets, qui consiste à identifier et localiser les objets présents dans une image, et la segmentation, dont l'objectif est de déterminer le contour précis d'un objet, sont des problèmes classiques en vision par ordinateur. Dans les dernières années, des avancées majeures ont cependant été réalisées pour ces problèmes grâce au développement d'algorithmes à base d'apprentissage automatique, tels que les réseaux de neurones convolutifs (CNN) profonds. Pour la détection d'objets, des architectures de réseaux comme Faster R-CNN (Faster Regional

Figure 0.1 Panne électrique résultant de forts vents au Québec.
Tirée de Hydro-Québec (2019, p. 1)

CNN – version améliorée du réseau R-CNN initialement proposé par Girshick et al.) (Ren, He, Girshick & Sun, 2015), et Single Shot Detection (SSD) (Liu,W. & Berg, 2016) ont permis des gains très importants de performance, dépassant même dans certains cas la précision humaine. De nombreuses architectures à base de CNN ont également été proposées pour la segmentation d'images, par exemple le réseau SegNet (Segmentation Network) (Badrinarayanan, Kendall & Cipolla, 2017), offrant dans plusieurs cas une précision comparable à celle d'humains. Cependant, une limitation importante de ces approches est de ne pas séparer les objets appartenant à une même classe (e.g., deux voitures dans une même image). Pour pallier ce problème, des réseaux de segmentation d'instances comme Mask R-CNN (Mask Regional CNN) (He, Gkioxari, Dollár & Girshick, 2017), ont récemment été proposés. Typiquement, ces architectures ajoutent à un réseau pour la détection d'objets une seconde branche qui prédit pour chaque proposition d'objet et chaque classe un masque de segmentation. La région segmentée correspond alors au masque de la classe prédite par le réseau de détection.

Bien que les réseaux profonds aient permis d'améliorer considérablement la détection et la segmentation d'objets dans les images, l'entraînement de ces réseaux requiert fréquemment un grand nombre d'exemples annotés. Or, dans de nombreux cas, de telles annotations ne sont pas disponibles et leur acquisition représente une tâche à la fois complexe et coûteuse. À titre d'exemple, la segmentation manuelle des objets dans une image peut prendre plusieurs minutes, et des centaines d'images sont souvent nécessaires pour un bon entraînement. Pour remédier à ce problème, plusieurs travaux récents se sont donc concentrés sur le développement de méthodes à base d'apprentissage semi-supervisé pour la détection (Bilen & Vedaldi, 2015 ; Cinbis, Verbeek & Schmid, 2017 ; Tang, Wang, Gao, Dellandréa, Gaizauskas & Chen, 2016 ; Yan, Liang, Pan, Li & Zhang, 2017) et la segmentation (Bai,W. & Rueckert, 2017 ; Hung, Tsai, Liou, Lin & Yang, 2018 ; Souly, Spampinato & Shah, 2017 ; Zhou, Wang, Tang, Bai, Shen, Fishman & Yuille, 2018) d'objets. Le principe commun de ces méthodes est d'exploiter la disponibilité de nombreuses images sans annotation ou avec des annotations partielles pour améliorer l'entraînement du modèle. Les approches semi-supervisées pour la détection d'objets reposent sur différentes techniques comme l'apprentissage par transfert d'un réseau de classification (Tang *et al.*, 2016), (Bilen & Vedaldi, 2015), l'auto-apprentissage avec l'algorithme Expectation-Maximization (EM) (Yan *et al.*, 2017) et l'apprentissage par instances multiples (Multiple instance learning – MIL) (Cinbis *et al.*, 2017) visant à assigner une étiquette à un ensemble d'exemples. La plupart de ces approches requièrent des étiquettes de classe pour chaque image, spécifiant si un objet d'une certaine classe est présent ou non dans l'image. Dans l'application ciblée par le projet, le nombre de classes est très limité (e.g., deux classes : poteau et transformateur), et ces classes sont la plupart du temps observées en même temps. Conséquemment, ces approches sont mal adaptées à notre problème. Étant un problème plus structuré que la détection (e.g., la taille de la sortie est connue), la segmentation d'images a donné lieu à un plus vaste éventail de techniques semi-supervisées, incluant l'auto-apprentissage (Bai & Rueckert, 2017), la distillation de connaissances entre des modèles enseignants et

élèves (Zhou *et al.*, 2018) et l'apprentissage adversaire (Souly *et al.*, 2017), (Hung *et al.*, 2018). Les approches à base d'apprentissage adversaire ont démontré un particulièrement grand potentiel pour la segmentation semi-supervisée. Typiquement, ces approches combinent un réseau générateur ayant pour but de générer des segmentations similaires à la vérité terrain, et un réseau discriminateur qui tente de déterminer si la segmentation est générée ou non. En entraînant ces deux réseaux de manière opposée (i.e., le générateur tente de maximiser l'erreur du discriminateur), on peut ainsi obtenir des segmentations plausibles pour les images non-annotées. Malgré son énorme potentiel, l'application de cette approche au problème de segmentation d'instances (i.e., détection combinée avec segmentation d'objets) reste à ce jour peu explorée.

Les travaux présentés dans ce mémoire ont pour objectif d'améliorer le processus de maintenance des équipements du réseau de distribution d'électricité d'Hydro-Québec en automatisant certains processus. Pour atteindre cet objectif, 5 contributions sont proposées :

- La détection et segmentation des composants attachés à un poteau utilitaire.
- La lecture du texte sur les plaques LCLCL et la correction des coordonnées géo-référencées.
- La prise de mesure de l'angle du poteau à partir de la segmentation.
- La création d'une application Android qui exploite les points précédents.
- La segmentation automatique des équipements à l'aide de réseaux de neurones adversaires.

La structure de ce mémoire est la suivante. Au premier chapitre, dix articles sont passés en revu dans le but de soutenir la recherche, dont la méthodologie est décrite au second chapitre. Au troisième chapitre, les résultats de la recherche sont exposés pour qu'aux prochains chapitres, quatre et cinq, les résultats y soit interprétés et discutés.

CHAPITRE 1

REVUE DE LITTÉRATURE

Ce travail de recherche repose sur différents concepts de l'apprentissage profond et la vision par ordinateurs introduits dans les dernières années. L'apprentissage adversaire permet de résoudre des problématiques concernant la taille du jeu de données en générant de nouvelles données synthétiques, avec un complément de *transfer learning* (Inoue, Chaudhury, Magistris & Dasgupta, 2018) entre les données réelles et synthétiques pour réduire la taille du domaine.

Chaînés à ce processus, des réseaux de segmentation par instances vont permettre d'isoler les sujets principaux dans les images générées. Des transformateurs spatiaux (Jaderberg, Simonyan, Zisserman & kavukcuoglu, 2015) sont utilisés pour rectifier le contenu de l'image pour effectuer des opérations, telles que de la reconnaissance textuelle dans une scène.

Les prochaines sous-sections présentent des travaux en lien avec ces concepts et leurs utilisations pour la recherche.

1.1 Synthétisation d'images réelles

L'obtention d'un ensemble volumineux de données réelles étiquetées est souvent difficile et dispendieux. Les données synthétiques sont plus faciles à obtenir et ajoutent d'autres avantages (Nikolenko, 2019), tel que l'automatisation des étiquettes. Les auto-encodeurs profonds (Kingma & Welling, 2013) offrent des fonctionnalités permettant de combiner le réel et le synthétique.

Malgré la quantité de données disponibles, certains éléments à identifier se retrouvent en quantités insuffisantes. Ces éléments, plus particulièrement les LCLCL, se retrouvent en moins grand nombre dans le jeu de données d'Hydro-Québec et construire un nouveau jeu de données est trop dispendieux et requiert trop de temps. L'attention se tourne donc vers un jeu de données synthétiques.

Dans (Inoue *et al.*, 2018), les chercheurs font face à un problème similaire où la quantité de données réelles pour faire l'entraînement de leur modèle est très limité. Le jeu de données synthétiques devient une solution intéressante pour l'entraînement. Cependant, dans les scénarios réels, la détection devient très mauvaise. Afin de faire un rapprochement entre ces deux types de données, un réseau de conversion à l'aide d'auto-encodeur et auto-décodeur a été mis en place pour synthétiser les images réelles lors de la détection.

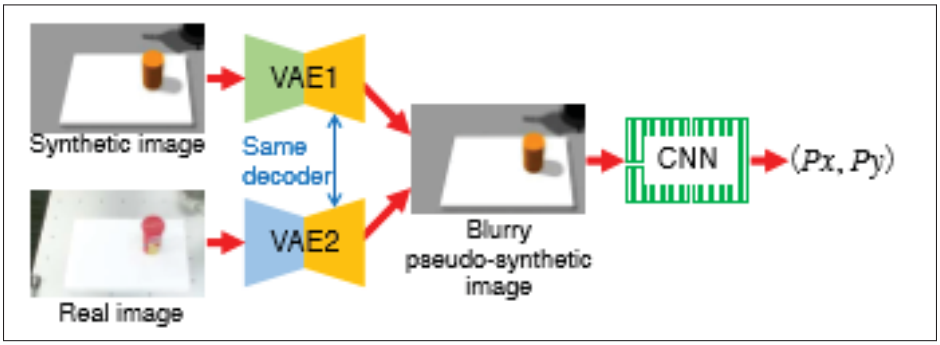

Figure 1.1 Architecture de synthétisation des images. Tirée de Inoue *et al.* (2018, p. 1)

Le principe derrière les auto-encodeurs (AE) peut être comparé au fonctionnement de compression et décompression de données. L'AE encode les données en entrée sous un format X où les poids des neurones agissent comme variables pour encoder les données pour, par la suite, faire l'opération inverse. Les portions encodeur et décodeur de l'AE peuvent être indépendantes l'une de l'autre et c'est la stratégie utilisée par la solution. Plus précisément, la solution utilise un auto-encodeur variationnel (VAE) qui exploite l'espace latent et les paramètres pour reconstruire les données, à l'opposé des données "compressées" de l'AE.

Dans un premier temps, le VAE est entraîné exclusivement avec des données synthétiques. Une fois ces couches bien entraînées, la portion encodeur est remplacée par un nouvel encodeur et est entraînée avec des images synthétiques alors que la partie décodeur est conservée telle quelle (les poids et couches sont figés). Dans un deuxième temps, le nouvel encodeur est entraîné avec des images réelles. Cette combinaison résulte en une pseudo-synthétisation des données en entrée lorsqu'elles sont décodées.

1.2 Localisation de texte synthétique dans des images réelles

Maintenant qu'il est possible d'utiliser des données synthétiques pour entrainer le modèle de détection (et de lecture) des LCLCL, il faut générer des images avec des plaques LCLCL synthétiques insérées dans des images réelles.

La méthode proposée dans (Gupta, Vedaldi & Zisserman, 2016) combine d'une part un engin de génération et de superposition naturelle de texte dans une scène et d'une autre part un modèle de détection de texte entrainé à partir des images avec le texte artificiel.

Figure 1.2 Texte synthétique dans une scène. Tirée de Gupta *et al.* (2016, p. 3)

Produire du texte synthétique réaliste dans une scène, de façon entièrement automatisée pour générer une grande quantité de données, requiert plusieurs étapes. Au début du processus, l'image est segmentée en plusieurs régions continues, selon la couleur et la texture, à l'aide de l'algorithme *graph-cut*. Par la suite à l'aide d'un réseau convolutif, une carte des profondeurs est générée pour ces régions. Suite à ces étapes, un texte est sélectionné aléatoirement depuis une banque de texte et, à l'aide de l'algorithme d'édition d'image de Poisson (Pérez, Gangnet & Blake, 2003), ce texte est intégré à l'image.

Pour une image naturelle, le texte est généralement imprimé sur la surface d'un objet. Dans le but de reproduire cet effet, certaines régions segmentées sont éliminées en utilisant l'algorithme RANSAC. RANSAC consiste en une méthode itérative où les données aberrantes sont éliminées selon une marge d'acceptation pour obtenir un résultat concluant.

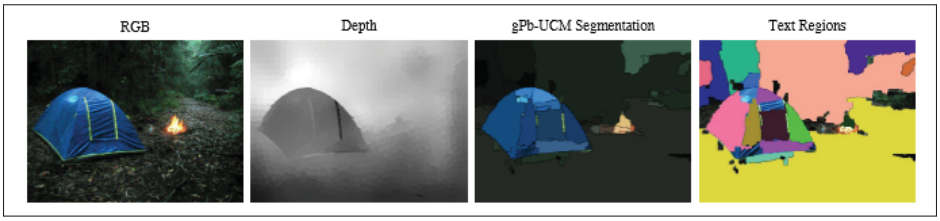

Figure 1.3 Différentes représentation de l'image au courant du processus. Tirée de Gupta *et al.* (2016, p. 3)

1.3 Réseaux de neurones profonds pour la reconnaissance de séquences de caractères

Les plaques LCLCL synthétiques permettent d'entraîner le modèle sur une quantité quasi-illimitée de scénarios où les plaques ont été apposées. Avec ce nouveau jeu de données prêt à être utilisé, il devient possible de s'attaquer à la problématique de lire la chaîne de caractère qui compose une plaque LCLCL, soit une chaîne de caractères verticale.

L'étude dans (Shi, Bai & Yao, 2017) explore le problème de la reconnaissance de texte dans une scène, l'un des problèmes les plus difficiles dans le domaine de la reconnaissance d'image. La solution proposée par l'étude se base sur une architecture de réseau de neurones composée de couches d'extraction de caractéristiques, de modélisation de séquences et de transcription du texte. Ce modèle offre une robustesse à l'échelle (taille des objets) et est léger, ce qui est important à la résolution de notre problématique.

Le réseau débute avec une séquence de couches de convolutions pour extraire les caractéristiques de l'image utilisée en entrée. Ces couches représentent le cœur des réseaux de convolution et sont la source de la puissance de ces réseaux. Pour donner suite aux couches de convolution, des vecteurs de caractéristiques sont produits à l'aide des caractéristiques extraites précédemment. Ces vecteurs permettent d'identifier les champs réceptifs de l'image et faire une association avec les caractéristiques. Un réseau profond bidirectionnel composé de LSTM (Long Short Term Memory, Mémoire long-court terme) fait par la suite l'identification des étiquettes. Chaque module LSTM est connecté avec celui qui le précède dans les deux directions de la séquence, ce qui permet à l'information de transiger aisément et contribue à l'apprentissage des

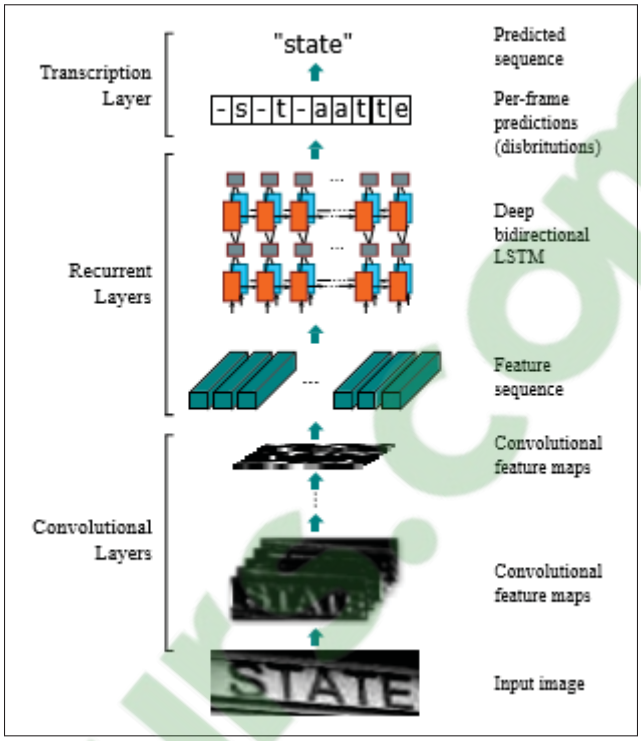

Figure 1.4 Structure du réseau neurones.
Tirée de Shi *et al.* (2017, p. 2)

séquences des étiquettes. Finalement, une couche de transcription prédit la séquence de texte à l'aide d'une activation CTC (Connectionist Temporal Classification, Classification temporelle connexionniste). Le CTC utilise la carte de probabilité produite par le réseau récurrent LSTM pour définir le mot en ignorant les caractères "blanc" (-) et les répétitions.

1.4 Restauration d'images endommagées à l'aide de réseaux profonds

Étant exposées aux intempéries du climat québécois, certaines plaques signalétiques LCLCL ont été endommagées au fil du temps, rendant la lecture de certains caractères parfois difficile, même pour un être humain. Ainsi, (Ulyanov, Vedaldi & Lempitsky, 2017) propose plusieurs solutions à cette problématique. Le réseau de neurones profond proposé permet, de façon générique, de réduire le bruit et de reconstruire des images avec lesquelles il n'a pas été entraîné, en utilisant les données statistiques et caractéristiques des images.

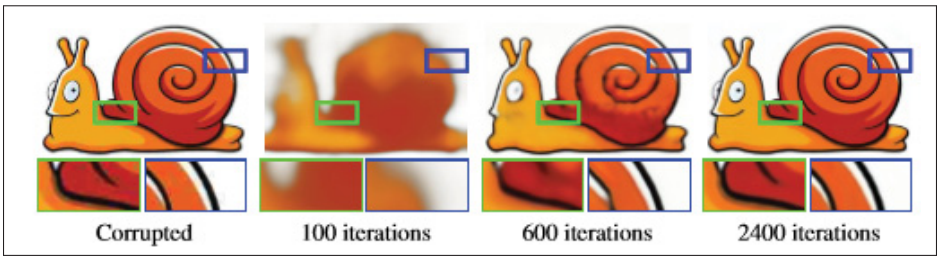

Figure 1.5 Reconstruction partielle d'une image. Tirée de Ulyanov *et al.* (2017, p. 4)

La solution choisie utilise un réseau de neurones profond pour la génération d'images, plus particulièrement un réseau U-Net en forme de sablier avec des raccourcis de connexion. Ce réseau interprète la problématique selon un problème de paramétrisation où ces paramètres apprennent à produire des opérations inverses, telles que la réduction de bruit, l'augmentation de résolution et la reconstitution d'images.

Pour la réduction de bruit et la reconstitution d'images, le réseau est entraîné avec des images "corrompues" en entrée et des images parfaites comme résultat désiré. Le tout suit la formule $E(\mathbf{x}; \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|$ où $\mathbf{x}$ est l'image en entrée et $\mathbf{x}_0$ l'image désirée, ce qui donne le problème d'optimisation $\min_\theta \|\mathbf{f}_0(z) - \mathbf{x}_0\|^2$, θ étant la valeur de paramètre recherchée (z est l'espace spatial du tenseur). En minimisant cette variable, le réseau s'adapte à la permutation des pixels entre les images et apprend à corriger les défauts.

1.5 Ré-alignement 3D d'objets dans un espace 2D

Une autre problématique est le degré de liberté quasi-illimité de prise de vue de l'appareil mobile. Avec une multitude de possibilités pour une simple image, la quantité de données nécessaires pour faire un entraînement adéquat du réseau de neurones s'accroît de manière importante.

La solution proposée par l'équipe de DeepMind dans (Jaderberg *et al.*, 2015) permet d'éviter ce scénario grâce à l'utilisation d'un module de transformation spatiale. Ce module permet de transformer dans l'espace spatial les caractéristiques de l'image et ainsi rendre le modèle spatialement invariant aux translations, à l'échelle, aux rotations et aux déformations de l'image.

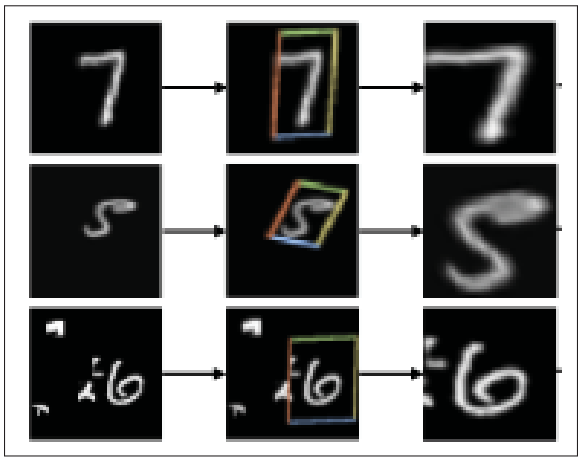

Figure 1.6 Transformation spatiale
pour corriger la position du caractère.
Tirée de Jaderberg *et al.* (2015, p. 2)

La transformation spatiale montrée dans la figure 1.7, est formulée comme suit :

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1.1}$$

où (x_i^s, y_i^s) sont les coordonnées source, (x_i^t, y_i^t) sont les coordonnées cibles et A_θ la matrice de transformation affine.

Il s'agit d'un module qui modifie les caractéristiques extraites par les couches de convolution en un seul passage. Cette transformation est conditionnée au type d'image utilisée en entrée, produisant ainsi un ensemble de caractéristiques uniques. Pour plusieurs canaux de l'image (ex. RGB), la même opération est effectuée sur chaque canaux.

La première opération détermine la tâche à effectuer sur l'ensemble des caractéristiques de l'image, et ce à partir d'un ensemble de couches cachées. La seconde opération établie une grille en échantillonnant un noyau moyen pour chaque position dans l'image. Cette grille représentera les pixels de l'image résultante à la fin du processus.

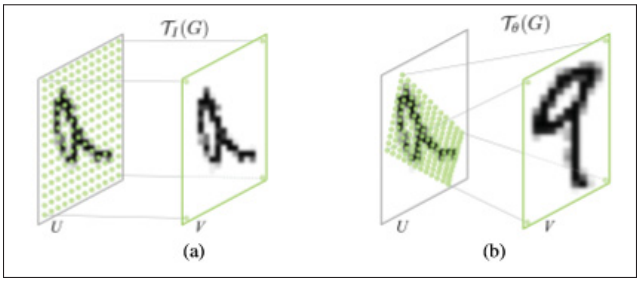

Figure 1.7 Création de la grille. Tirée de Jaderberg *et al.* (2015, p. 4)

Cette grille permet d'effectuer plusieurs types d'opérations de transformation à l'image, tel que mentionné précédemment. Ces opérations, dans le cas d'une opération de transformation affine, utilisent 6 paramètres (θ_{11} à θ_{23}) pour définir quel type de transformation sera appliqué. Il s'agit ici de transformations matricielles.

La troisième et dernière étape est l'application de la transformation sur chaque canal. L'échantillonnage étant fait de façon identique sur chaque canal, la transformation est donc elle aussi identique. Cette série d'opération est autonome et peut donc être utilisée dans n'importe quel type de réseau de convolution.

1.6 Apprentissage adversaire sur images synthétiques sans étiquettes

Les réseaux de génération de données par apprentissage adversaires (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville & Bengio, 2014) permettent d'halluciner des images similaires à celles utilisées lors de l'entraînement. Ainsi il est possible de générer une quantité quasi-illimitée de nouvelles données.

Une approche au problème du manque de données pour la segmentation est la génération d'un jeu de données entièrement synthétiques pour faire l'entraînement du modèle de réseau de neurones. En étant générées synthétiquement, les images posséderaient automatiquement les annotations nécessaires pour faire l'entraînement et, ainsi, l'étape de segmentation manuelle ne serait plus nécessaire. Cependant, tel que mentionné précédemment, utiliser des images

synthétiques en entraînement et images réelles durant les tests effectués ne donne typiquement pas de bons résultats.

Un problème activement recherché dans le domaine de l'imagerie et apprentissage machine est le rapprochement entre les données synthétiques et données réelles. La technique dans (Shrivastava, Pfister, Tuzel, Susskind, Wang & Webb, 2017) propose l'utilisation de réseaux adversaires pour créer une version raffinée des images synthétiques.

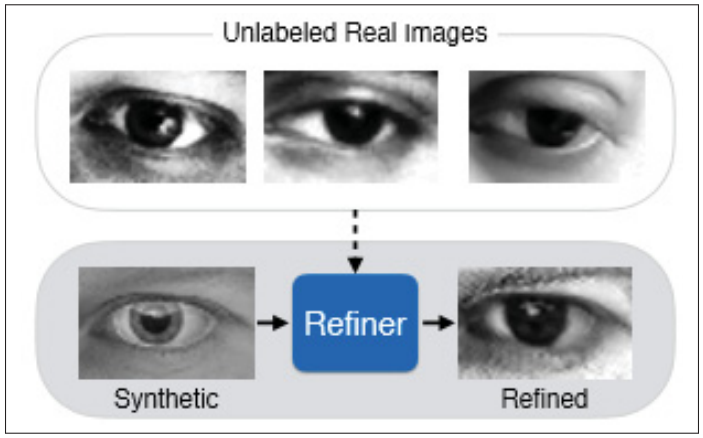

Figure 1.8 Raffinement des images synthétiques.
Tirée de Shrivastava *et al.* (2017, p. 1)

Cette technique repose sur l'utilisation de GANs (*Generative Adversarial Networks* – Réseaux Adversaires Génératifs). Comme illustré à la figure 1.8, ce type de réseau hallucine des images se rapprochant de la réalité, selon la variété des données utilisées pour l'entraînement. Cette approche ajoute une étape supplémentaire au processus de génération des images synthétiques, soit celle du raffinement. Ce dernier minimise la combinaison de la perte du réseau adversaire et ajoute une variable d'auto-régulation. La variable d'auto-régulation minimise les différences entre les images synthétiques et les images raffinées en réduisant la différence pixel-à-pixel entre les deux types d'images selon la formule $\ell_{reg} = \|\psi(\overline{x}) - x\|$, où ψ est le lien entre l'image et ses caractéristiques, $\overline{x}$ est l'image raffinée et x l'image synthétique.

Tel que montré à la figure 1.9, le résultat est par la suite transféré au discriminant qui classifie les images selon deux types d'étiquette : Réelle ou Synthétique. Avec l'étiquette choisie, le

Figure 1.9 Discriminant et probabilités.
Tirée de Shrivastava *et al.* (2017, p. 4)

réseau compare sa décision avec la vérité terrain et propage la perte sur son propre réseau et le réseau générateur, ce qui résulte à la fois en de meilleurs libellés ainsi qu'à de meilleures images synthétiques.

1.7 Réseaux adversaires pour la segmentation sémantique

La quantité de données dont Hydro-Québec dispose est énorme, soit près de 500,000 images. Cependant, ces images ne sont pas segmentées et cela représente une étape qui requiert beaucoup de temps. Dans le but d'optimiser les efforts de recherche, une approche de segmentation par réseau adversaire est mise de l'avant pour automatiser la segmentation des images du jeu de données.

Dans (Luc, Couprie, Chintala & Verbeek, 2016), les auteurs suggèrent une approche générative où un réseau est entraîné pour accomplir des segmentations et un autre réseau est entraîné pour discriminer entre les segmentations générées automatiquement et les segmentations effectuées manuellement.

La solution propose un modèle hybride à double perte. La première perte consiste en une entropie croisée sur les étiquettes de chaque pixel composant l'image où les étiquettes sont indépendantes. Ce type de perte est typique dans les réseaux de segmentation. La deuxième perte consiste en un signal provenant du réseau discriminant selon la prédiction de ce réseau,

soit étant correcte ou incorrecte :

$$\sum_{n=1}^{N} \ell_{mce}\big(s(\mathbf{x}_n), \mathbf{y}_n\big) + \lambda \ell_{bce}\big(a(\mathbf{x}_n, s(\mathbf{x}_n)), 1\big) \tag{1.2}$$

où $\sum_{n=1}^{N} \ell_{mce}\big(s(\mathbf{x}_n), \mathbf{y}_n\big)$ représente la somme de la perte de l'entropie croisée moyenne entre une prédiction $s(\mathbf{x}_n)$ et la vérité $\mathbf{y}_n$. $\lambda \ell_{bce}$ est la perte de l'entropie croisée binaire sur $a(\cdot)$ qui est la prédiction du réseau adversaire.

1.8 Réseaux adversaires profonds pour la segmentation d'images sans étiquettes

Pour augmenter la robustesse de la segmentation par réseau adversaire, la solution dans (Zhang, Yang, Chen, Fredericksen, Hughes & Chen, 2017) propose de comparer et potentiellement combiner le résultat des solutions afin d'obtenir des segmentations de hautes qualités. Un réseau adversaire profond utilise des images annotées et non-annotées pour segmenter les images non-annotées à l'aide d'un réseau de neurones pour segmenter. Ce dernier permet aussi de segmenter alors qu'un autre réseau de neurones permet d'évaluer la qualité des segmentations (discriminant).

Deux réseaux sont impliqués dans le modèle proposé : un réseau de segmentation et un réseau d'évaluation (discriminant). Le réseau de segmentation génère des segmentations probabilistes correspondant à l'image en entrée et le réseau d'évaluation évalue les segmentations en leur donnant un score (1 pour bon, 0 pour mauvais). Durant l'entraînement, le réseau d'évaluation donne des scores plus élevés aux données annotées et moins élevés aux données non-annotées volontairement dans le but de forcer le réseau de segmentation à générer des segmentations de meilleure qualité. Le réseau de segmentation cherche donc à tromper le réseau d'évaluation.

1.9 Apprentissage adversaire pour segmentation semi-supervisée

Malgré l'utilisation de réseaux adversaires pour accélérer la segmentation des images, la quantité d'efforts nécessaires surpasse le temps assigné au projet. La quantité raisonnable d'images segmentées pour effectuer l'entraînement adversaire reste trop grande.

Alors que la majorité des discriminants entraînés dans les réseaux adversaires sont capables de définir si la classe est vraie ou artificielle, la méthode proposée dans (Hung *et al.*, 2018) ajoute des signaux supplémentaires lors du calcul de la perte du réseau adversaire. Cela permet ainsi de propager l'erreur et ajouter de l'information sur les masques de segmentations générés et leur vérité.

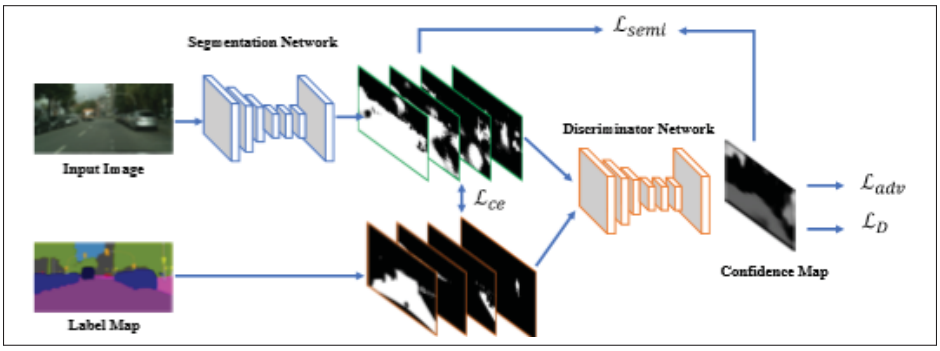

Figure 1.10 Aperçu du système de segmentation semi-supervisé. Tirée de Hung *et al.* (2018, p. 4)

Cette approche est illustrée à la figure 1.10. Un réseau génère des segmentations et un autre réseau juge si les segmentations qui lui sont fournies proviennent des segmentations manuelles ou des segmentations automatiques. Le réseau de segmentation utilisé est DeepLab-V2 avec ResNet-101 et des poids préinitialisés provenant de ImageNet et MSCOCO. Le réseau discriminant consiste en cinq couches de convolutions utilisant chacune une activation "Leaky-ReLU" (Rectified Linear Unit, Unité linéaire rectifiée).

La fonction de coût du réseau discriminant compare pour chaque portion de l'espace vectoriel de l'image (hauteur $\times$ largeur $\times$ canaux) si la prédiction sur le type de segmentation (manuelle,

automatique) est correcte et par la suite propage l'erreur dans son propre réseau et ultérieurement partage l'information au réseau de segmentation.

Le réseau de segmentation utilise la fonction de coût multi-tâche (semi-supervisé + discriminant) suivante pour renforcer la qualité de ses segmentations :

$$\mathcal{L}_{seg} = \mathcal{L}_{ce} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{semi}\mathcal{L}_{semi} \tag{1.3}$$

La perte du réseau adversaire ($\mathcal{L}_{adv}$) indique au réseau de segmentation si la segmentation effectuée trompe ou non le réseau adversaire. Cette valeur influencera le comportement du réseau de segmentation au cours de l'entraînement (discriminant trompé = peu de changements ; discriminant détecte le type de segmentation correctement = plus de changements). La perte des données annotées ($\mathcal{L}_{ce}$), définie comme

$$\mathcal{L}_{ce} = -\sum_{h,w}\sum_{c\in C} Y_n^{h,w,c} \log\left(S(\mathbf{X}_n)^{(h,w,c)}\right) \tag{1.4}$$

indique si les segmentations générées sont près de la vérité. Y_n indique la vérité et $S(\mathbf{X}_n)$ le résultat de la prédiction.

La fonction $\mathcal{L}_{ce}$ calcule la perte à l'aide des données étiquetées dans le but d'aider à diriger le gradient et permettre au réseau générateur d'apprendre plus rapidement. Pour les données non-étiquetées, $\mathcal{L}_{ce}$ n'est pas utilisée. $\mathcal{L}_{adv}$ est la perte adversaire considérant un réseau discriminant convolutif, et est définie comme

$$\mathcal{L}_{adv} = -\sum_{h,w} \log\left(D(S(\mathbf{X}_n))^{(h,w,c)}\right). \tag{1.5}$$

L'élément clé du réseau est sa capacité à effectuer un entraînement avec des données non-annotées. La perte sur les données annotées et non-annotées ($\mathcal{L}_{semi}$),

$$\mathcal{L}_{semi} = -\sum_{h,w}\sum_{c\in C} Y_n^{h,w,c} I\left(D(S(\mathbf{X}_n))^{(h,w)} > T_{semi}\right) \cdot \hat{Y}_n^{(h,w,c)} \log\left(S(\mathbf{X}_n)\right)^{(h,w,c)}\Big), \tag{1.6}$$

met en évidence les régions de confiance à l'aide du discriminant entraîné sur les données non-annotées et confirmé avec les données annotées, en comparant ces régions avec les annotations. Ainsi chaque portion du réseau contribue à l'amélioration des segmentations.

1.10 Apprentissage mutuel profond

La mobilité et la rapidité du modèle sont des contraintes prioritaires dans la réalisation du projet. Dans l'esprit de ne pas limiter les possibilités et options lors de la réalisation du réseau de neurones principal, une autre version du réseau de neurones principal sera développée et entraînée de façon adjacente au réseau principal.

L'apprentissage profond mutuel proposé dans (Zhang, Xiang, Hospedales & Lu, 2018) repose sur le modèle "professeur-élève". Ce modèle, illustré à la figure 1.11, consiste en un apprentissage collaboratif entre plusieurs modèles où les connaissances sont partagées entre les élèves et le professeur. Cette collaboration résulte en un gain de performances pour tous les modèles de réseau de neurones impliqués.

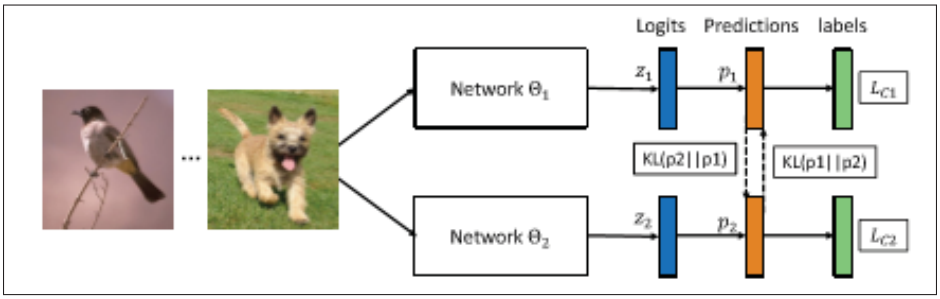

Figure 1.11 Apprentissage mutuel. Tirée de Zhang *et al.* (2018, p. 3)

Les réseaux de neurones profonds modernes offrent des performances selon l'état de l'art, mais souvent au détriment d'être soit large, profond ou alors lent à l'utilisation. Ces éléments impactent la portabilité du réseau et potentiellement sa rapidité. L'entraînement profond mutuel rend possible le partage des acquis lors de l'entraînement de plusieurs réseaux, ce qui résulte en des réseaux de structures différentes, mais de performances similaires. La fonction de coût globale

est altérée au même moment où la perte du modèle calcule la distance entre les prédictions des réseaux impliqués :

$$D_{KL}(\mathbf{p}_2 \| \mathbf{p}_1) \;=\; \sum_{i=1}^{N} \sum_{m=1}^{M} \mathbf{p}_2^m(\mathbf{x}_i) \log \frac{\mathbf{p}_2^m(\mathbf{x}_i)}{\mathbf{p}_1^m(\mathbf{x}_i)} \tag{1.7}$$

La divergence de Kullback-Leibler (D_{KL}), (Kullback & Leibler, 1951), permet de mesurer la compatibilité entre les prédictions des réseaux en mesurant la dissimilarité entre deux distributions de probabilités avec les termes $\mathbf{p}_1$ et $\mathbf{p}_2$, où $\mathbf{p}_x$ indique la prédiction d'un réseau x. Cette divergence est utilisée pour calculer la perte totale de chaque réseau, L_{θ_1} et L_{θ_2}, comme suit :

$$L_{\theta_1} \;=\; L_{C_1} \;+\; D_{KL}(\mathbf{p}_2 \| \mathbf{p}_1) \tag{1.8}$$

$$L_{\theta_2} \;=\; L_{C_2} \;+\; D_{KL}(\mathbf{p}_1 \| \mathbf{p}_2) \tag{1.9}$$

Ce partage améliore l'apprentissage des réseaux et améliore aussi les performances finales lors de l'utilisation du réseau.

1.11 Conclusion

Les avancées et découvertes de ces travaux, parfois spectaculaires, ne sont que le début d'une nouvelle ère de recherche, propulsé par les réseaux profonds.

Par contre, ces travaux comportent certaines limitations ou incompatibilités avec la recherche. La reconnaissance textuelle nécessite d'avoir au préalable un jeu de données volumineux pour l'entraînement. Les objets synthétisés dans ces travaux sont dans certains cas simples ou leurs origines proviennent d'un environnement contrôlé.

Néanmoins, ces travaux et tout ceux sur lesquels ceux-ci reposent vont permettent à la recherche d'avancer, souvent via la combinaison de plusieurs travaux afin de répondre à une problématique bien spécifique.

CHAPITRE 2

MÉTHODOLOGIE

2.1 LCLCL synthétiques

La lecture des plaques d'identification utilisées par Hydro-Québec, les plaques LCLCL, représentent le premier défi à conquérir. Le texte de ces plaques doit être lu en temps quasi-réel afin de pouvoir confirmer l'équipement présent sur le poteau utilitaire et d'effectuer une vérification des coordonnées géographiques dans la base de données.

Cependant Hydro-Québec ne possède pas un inventaire d'images annotées pour les plaques LCLCL et il existe présentement plus de 1.1 millions de plaques en circulation, distribuées à la grandeur du Québec.

Avant de statuer sur l'utilisation d'un jeu de données synthétiques, plusieurs plans ont été mis de l'avant afin de créer un jeu de données représentatif pour effectuer l'entraînement du premier modèle de détection de texte.

2.1.1 Données

Le tout premier plan fut naturellement de créer un jeu de données avec des images réelles pour faire l'entraînement. Hydro-Québec possède une banque d'images qui a été créée durant l'inspection de l'état de santé des poteaux du réseau électrique. Cette banque d'images possède par contre seulement une quantité limitée d'images de poteaux portant une plaque LCLCL visible et donc, ce plan a été abandonné. Par la suite, une approche avec des plaques synthétiques qui répliquent les caractéristiques de base d'une plaque LCLCL, sans prendre en compte la diversité et complexité de l'environnement où la plaque est posée, a débuté.

Après des résultats infructueux avec le modèle, une approche semi-synthétique a été considérée afin de combiner la qualité des images réelles et la facilité d'obtenir un grand jeu de données que les images synthétiques offrent. Une plaque LCLCL semi-synthétique résulte d'une combinaison

de plusieurs éléments réels, afin de créer des plaques fictives. Il s'agit ici de combiner des lettres extraites de plusieurs sources et de les combiner sur un arrière-plan sans lettre et d'ajuster le positionnement des lettres pour qu'elles soient bien alignées sur la plaque. Ainsi, à partir d'un échantillon de données, il est possible de générer une grande quantité de plaques.

Cette option n'a jamais vu le jour dû à la grande complexité de créer un logiciel qui va permettre de faire un assemblage automatique des composantes et donner un résultat comparable à une plaque LCLCL réelle. De plus, encore une fois, la collecte des données se présente comme une étape longue et ardue. C'est pour ces raisons que ce plan fût abandonné.

De retour aux plaques synthétiques, où l'objectif est d'éliminer l'écart entre le synthétique et le réel en utilisant des techniques de manipulation d'images mixtes et aléatoires. Cette nouvelle approche a permis d'obtenir les premières prédictions correctes sur le jeu de données réelles. Parmi les techniques utilisées, les principaux opérateurs sont *skew* et *projection*. Le premier sert à recréer un semblant de rotation dans l'image, afin de simuler la non-perpendicularité de la plaque LCLCL face au sol. En d'autres mots, la plaque dans l'image ne sera pas toujours droite et bien alignée. La deuxième opération, essentielle au succès des prédictions, représente l'angle de perspective entre la caméra et la plaque sur le poteau. Une autre grande avancée est la "binarisation" de l'image, c'est à dire la conversion vers deux couleurs unies, qui permet d'éliminer la majorité du bruit et d'éviter des distractions potentielles lors de l'apprentissage. La binarisation s'effectue à l'aide des données HSV de l'image afin de calculer la luminosité moyenne (*v-mean*) de l'image et ainsi pour définir la borne limite entre la couleur A (noir) et la couleur B (blanc). L'avantage de cette technique est qu'elle normalise l'image tout en éliminant la majorité du bruit. Ce processus, résumé dans l'algorithme 2.1, est à la fois appliqué aux données d'entraînement, de validation et de test.

Algorithme 2.1 Algorithme de binarisation

Entré : *Image BGR dans le format HSV,* $\mathcal{I}_m = \{\boldsymbol{h, s, v}\}$
Sortie : *Image normalisé, noir et blanc,* $\mathcal{I}'_m = \{\boldsymbol{0, ..., 255}\}$*, par pixel*
Conversion en tons de gris, de 3 canaux à 1, $\mathcal{I}_{BGR} \rightarrow \mathcal{I}_G$
$\mathcal{I}_m \leftarrow \mathcal{I}_G$
for *Les pixels* $l_v \in \mathcal{I}_m$ *(*$l_v = 1, ..., N$*)* **do**
    for *Les valeurs de luminosité* $v \in \updownarrow_v$ *(*$l_v = 1, ..., 255$*)* **do**
        $lv_{sum} \leftarrow v$
    end
end
$v_{mean} \leftarrow \frac{lv_{sum}}{N} \cdot 0.85$
for *Les pixels* $l_v \in \mathcal{I}_m$ *(*$l_v = 1, ..., N$*)* **do**
    Trouver le seuil du maximum de luminosité
    if $l_v \geq v_{mean}$ **then**
        $l_v \leftarrow v_{mean}$
    end
end

Une caractéristique intéressante des images synthétiques est l'arrière-plan généré aléatoirement pour chaque plaque LCLCL. Cet arrière-plan, illustré à la figure 2.1, cherche à reproduire la texture du poteau de bois sur lequel les plaques LCLCL sont fixées. Puisque les poteaux varient en âge, type de bois utilisé et dommages infligés, plusieurs patrons ont été développés afin de couvrir autant de scénarios que possible.

Une étape cruciale a été franchise à partir de la 16$^{\text{ème}}$ itération, l'étape où les plaques synthétiques sont indiscernables des plaques réelles (voir la figure 2.2). Cette étape confirme que l'écart entre les plaques réelles et synthétiques est maintenant assez faible et que seuls les cas d'exceptions provenant des plaques réelles restent à être gérer (ex : Dommages, neige, poussière).

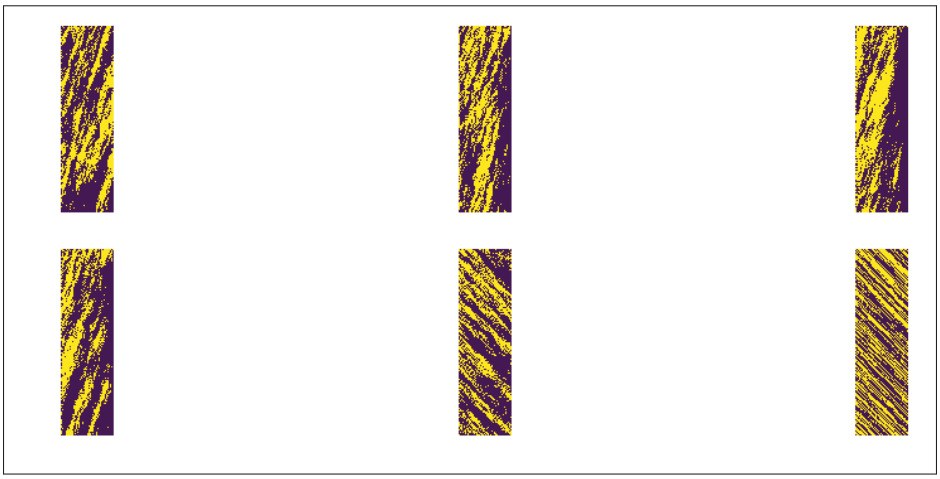

Figure 2.1 Texture simulée d'un poteau de bois

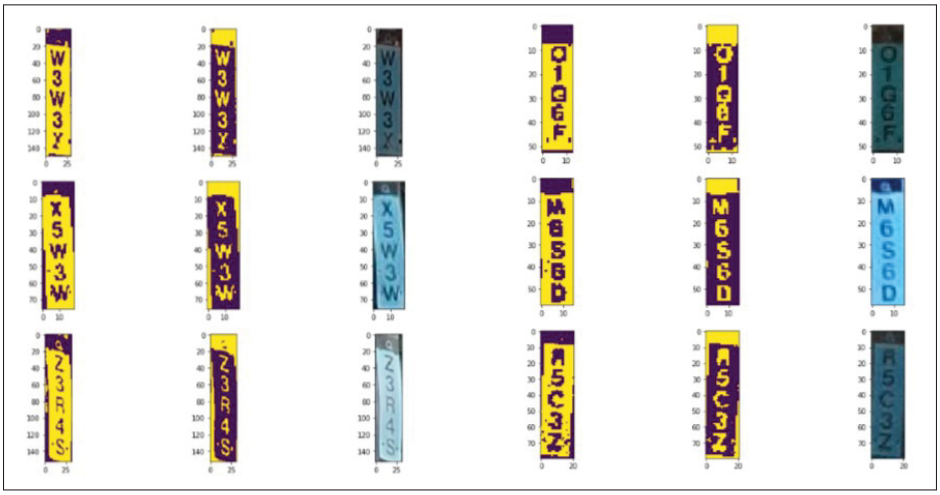

Figure 2.2 Mélange de données réelles et synthétiques

2.1.2 Modèles de lecture de textes verticaux

Deux modèles ont été employés pour la lecture de texte. Le premier est basé sur l'utilisation de couches de convolution et neurones récurrentes à portes (Convolution+GRU), nommé GRU-Net (voir la figure 2.3). Le deuxième réseau utilise comme base un réseau profond composé de couches de convulsion inter-connectées, appelé DenseNet (Huang, Liu, Van Der Maaten & Weinberger, 2017) (voir la figure 2.4).

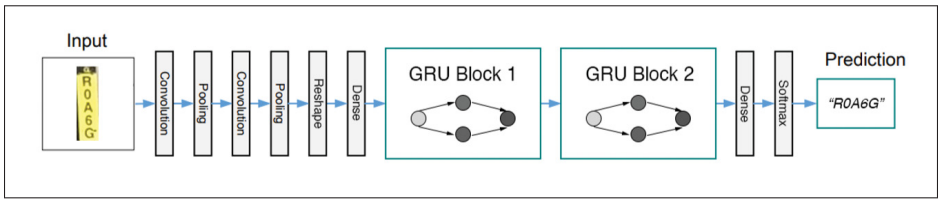

Figure 2.3 GRU-Net

GRU-Net utilise une combinaison de couches de convulsion, GRU (Cho, van Merrienboer, Bahdanau & Bengio, 2014) et une fonction de coût CTC (Graves, Fernández & Gomez, 2006) pour faire la prédiction du texte sur l'image. Les couches de convulsion font ressortir les caractéristiques de l'image suivi de deux blocs de couches GRU connectés séquentiellement, chacun composé de deux unités GRU en parallèles. Finalement, une couche de densification réduit les dimensions et une activation Softmax fait ressortir les prédictions.

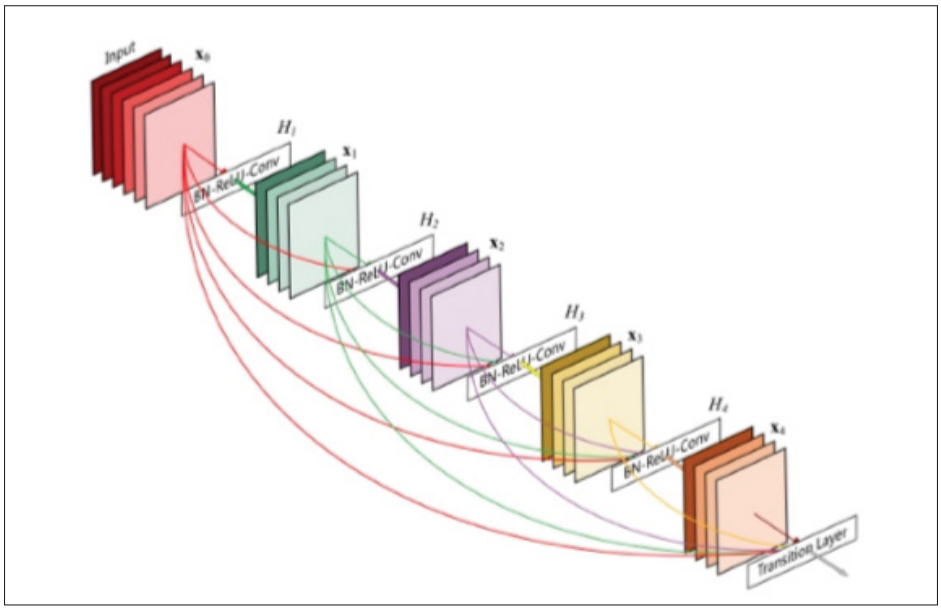

Figure 2.4 DenseNet. Tirée de Huang *et al.* (2017, p. 1)

Un aspect important de ce modèle, tout aussi applicable à DenseNet, est l'ajout d'une couche de synthétisation de l'image grâce à l'utilisation d'auto-encodeurs (voir figure 2.5).

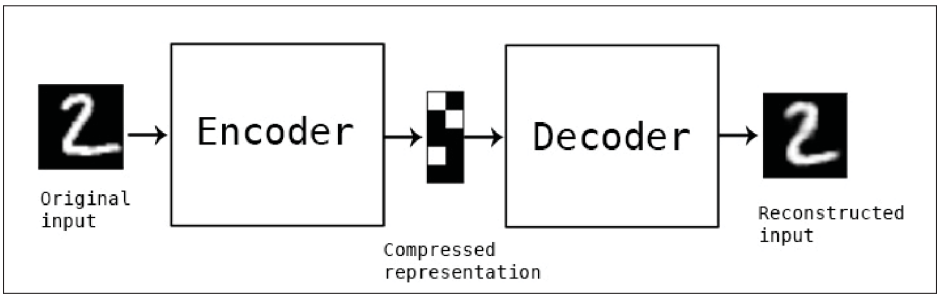

Figure 2.5 Auto-encodeur. Tirée de Chollet (2016, p. 1)

Un DAE est entrainé sur des images synthétiques pour recréer les images synthétiques avec un faible niveau de perte d'information contenu dans l'image. Une fois le réseau entraîné, la couche du décodeur est figée et la couche de l'encodeur est ré-entrainée, mais cette fois-ci en utilisant des images réelles. Cet auto-encodeur hybride permet de créer des versions synthétiques des images réelles (voir la figure 2.6).

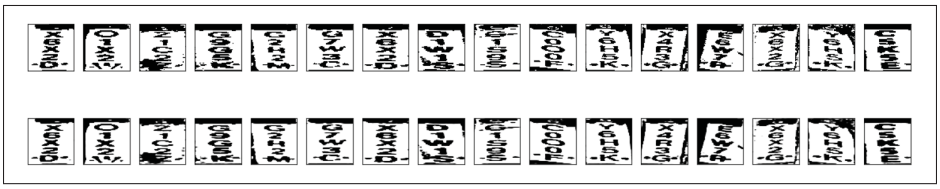

Figure 2.6 Synthétisation des images réelles

Le deuxième réseau sur lequel nous avons travaillé est basé sur l'implémentation Keras de DenseNet. Cette implémentation nous permet de modifier rapidement et facilement la profondeur du réseau. DenseNet peut être vu comme une avancée face aux réseaux de type Resnet, qui eux-mêmes étaient une avancée sur les réseaux de types Convnet. La grande particularité de DenseNet est que chaque bloc de convulsion prend en entrée l'entièreté des sorties des blocs similaires précédents. De plus, à la fin de chaque bloc, une normalisation (Batch Normalization) est utilisée pour ajuster et échelonner les activations et éviter la disparition du gradient. La fonction de coût reste la même que pour GRU-Net, soit la fonction CTC.

Pour chaque réseau, en plus des manipulations effectuées sur les données en entrée, des modifications supplémentaires ont été mises à l'essai afin d'augmenter le taux de réussite des

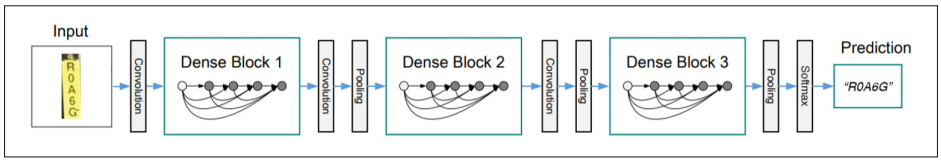

Figure 2.7 DenseNet, version modifiée

prédictions des réseaux. Plus spécifiquement, trois opérations non-exclusives ont été testées : redimensionnement des images avec couleurs, redimensionnement des images en tons de gris et inspiré du FSNS Dataset (Smith, Gu, Lee, Hu, Unnikrishnan, Ibarz, Arnoud & Lin, 2016) le chaînage de cinq images légèrement modifiées du même panneau LCLCL.

2.2 Classification et segmentation des poteaux et transformateurs

Le réseau de distribution électrique d'Hydro-Québec est d'une longueur totale de plusieurs centaines de milliers de kilomètres avec plus d'un million d'équipements attachés. Cette grande quantité d'équipements doit être entretenue périodiquement afin d'assurer une qualité de service aux consommateurs. Dans le but d'accélérer la maintenance des poteaux et transformateurs, un réseau de neurones spécialisé dans la segmentation de poteaux et transformateurs a été développé.

L'utilité de la segmentation est de pouvoir identifier et de sélectionner des poteaux et transformateurs dans une image afin d'obtenir leurs mesures. Ces informations seront utilisées par la suite afin de déterminer l'état d'usure et déterminer si ceux-ci doivent être réparés ou remplacés tout en indiquant quelle partie est impactée et la raison pour laquelle celle-ci est affectée.

2.2.1 Données

À partir de la banque d'images d'Hydro-Québec (environ un demi-million d'images), un échantillon de 300 images a été méticuleusement sélectionné afin d'être annotées. Les données sélectionnées représentent des cas réels qu'un employé d'Hydro-Québec risque de retrouver sur le terrain. Les éléments recherchés dans ces images sont : au minimum un poteau majoritairement

visible, avec ou sans transformateur(s) attaché(s), un minimum d'occlusions visuelles et, si possible, avec un LCLCL visible.

Ces données servent à tester plusieurs réseaux de neurones de segmentation dans le but d'identifier le plus performant.

2.2.2 Segmentation

Quatre réseaux de segmentations ont été mis à l'essai : Mask R-CNN, Deeplab, Segnet et RetinaNet + GrabCut. De ces réseaux, Mask R-CNN s'est avéré être le plus performant. Mask R-CNN est un réseau de neurones convolutif utilisant la détection d'objets par régions pour augmenter la qualité de ses segmentations. Produit des efforts de recherche de Facebook AI Research (FAIR), Mask R-CNN est un réseau utilisant des approches éprouvées et une combinaison ingénieuse de techniques de réseaux prédécesseurs, tels que Faster R-CNN (voir la figure2.8).

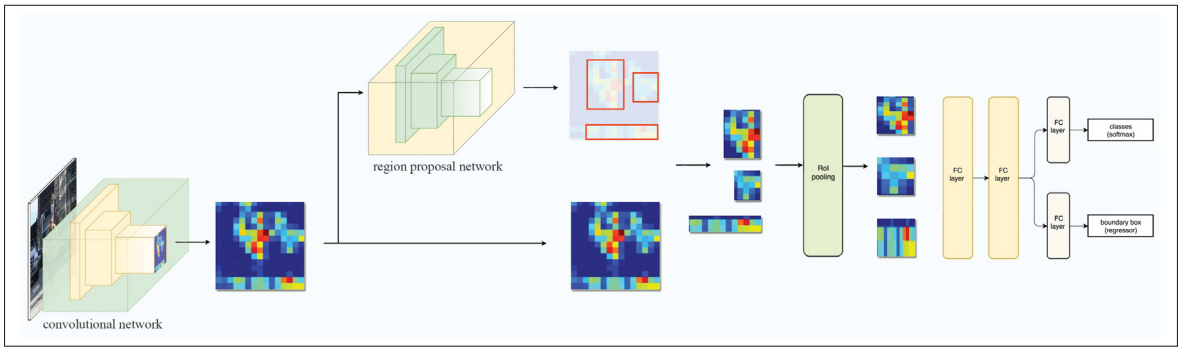

Figure 2.8 Faster R-CNN. Tirée de Hui (2018, p. 1)

Faster R-CNN joue un rôle crucial au sein du fonctionnement de Mask R-CNN, dont l'architecture est montrée à la figure 2.9. Faster R-CNN extrait les caractéristiques de l'image à l'aide d'un CNN. Par la suite, il utilise à nouveau un CNN spécialisé pour la proposition de régions pour créer des régions d'intérêts auprès de l'image (RoI, Region of Interest). Une transformation est appliquée à ces régions afin de les déformer vers des dimensions fixes. Finalement, ces régions

sont envoyées dans un réseau de couches totalement connectées (*fully connected*) pour faire la classification et la prédiction des boîtes de détection.

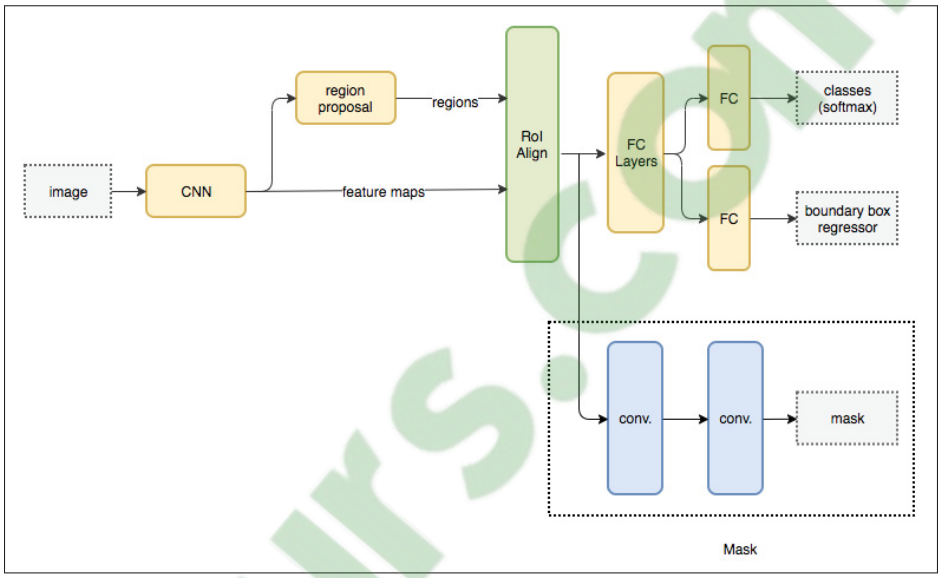

Figure 2.9 Mask R-CNN. Tirée de Hui (2018, p. 1)

Avec les régions d'intérêts en main, gracieuseté de Faster R-CNN, Mask R-CNN ajoute deux couches de convolutions pour extraire les masques de segmentation. Les améliorations de Mask R-CNN ne se limitent pas à ces couches de convolution. Mask R-CNN apporte aussi des améliorations auprès de la "couche" RoI en faisant un alignement des valeurs. Dans chaque groupe de régions, les valeurs sont échantillonnées par groupe de quatre en utilisant l'interpolation bilinéaire afin de générer des valeurs ajustées et ainsi éviter le désalignement du masque.

Le modèle de Mask R-CNN utilisé dans cette recherche est celui de Matterport [1] avec certains paramètres modifiés pour optimiser l'apprentissage sur les données. Les poids d'entraînement provenant de l'ensemble de données COCO ont été utilisés pour accélérer l'entraînement du modèle Mask R-CNN sur les données de l'IREQ. De cette façon, la qualité de l'entraînement a été améliorée et la durée d'entraînement requise pour obtenir des résultats cohérents réduite.

[1] Matterport Mask R-CNN github.com/matterport/Mask_RCNN

Tableau 2.1 Paramètres d'entraînement

Min. dimensions	Max. dimensions	Régions par image	Pas par époque
128	128	128	200

Les paramètres employés pour l'entraînement sont fournis dans le tableau 2.1. Cet entraînement s'effectue en deux temps. En premier, la tête du réseau qui renferme 24 couches est entraînée avec les données annotées pour préparer le réseau à la problématique à résoudre. Cette étape est de longue durée et est énormément influencée par les paramètres utilisés. Pour donner suite à l'entraînement de la tête de réseau, dans un deuxième temps, le réseau en entier, 236 couches, est entraîné pour faire des ajustements de précision.

2.2.3 Cotes d'états

La cote d'état des poteaux et transformateurs est établie à partir du résultat de la segmentation effectué sur l'image. La segmentation sur les transformateurs permet d'effectuer une classification entre un transformateur aillant une fuite d'huile et un transformateur sain. La segmentation sur les poteaux permet de calculer l'angle d'inclinaison de ces derniers. Le calcul de l'angle des poteaux, détaillé dans l'algorithme 2.2, s'effectue en 3 étapes : 1) Découpe de la segmentation en sous-sections 2) Calcul du centroïde des sous-sections 3) Calcul de l'angle à partir des centroïde.

La segmentation du poteau est découpée en en $N = 25$ sous-sections. La quantité N de sections a été déterminée à la suite d'essais et erreurs sur les données d'entraînement. Plus le nombre N est élevé, plus la précision de l'angle sera grande mais plus lent sera son calcul. Inversement, plus le nombre N est petit, moins précis sera l'angle mais le calcul sera plus rapide. $N = 25$ devient donc un compromis entre la précision et performance.

À chaque sous-section est attribuée une valeur unique, entre 1 et 255. Ces valeurs sont utilisées pour identifier de façon unique les sous-sections et permettre un affichage détaillé. Par la suite, à partir de ces valeurs uniques, la position matricielle de chaque pixel est extraite afin de calculer une moyenne en **X** et **Y** (voir la figure 2.10). Cette moyenne donne un centroïde approximatif,

Algorithme 2.2 Algorithme de calcul des sous-sections

Input : *Image mask with the region proposals*
Output : 25 *sub-sections, identified by a group identifier between* {***0, ... , 255***}
The area of the region proposal is divided in 25 groups of equal height, $\mathcal{I}_g$
$B_h \leftarrow$ Region height
for *all pixel group* $P_g \in \mathcal{I}_g$ *(*$I_g = 1, ..., 25$*)* **do**
    Group id $G_{id} \leftarrow (\frac{B_h - P_g id}{B_h} * 240) + 15$
    $P_g \leftarrow P_g[offset + P_g id, width] \cdot G_{id}$
    $Offset \leftarrow P_g id$
end

ce qui est idéal puisque cette estimation est versatile et résistante aux sous-groupe de formes non-rectangulaires. La technique pour calculer le centroïde est détaillée dans l'algorithme 2.3.

$$\begin{bmatrix} \cdots_{x,y} & \cdots_{x,y} & \cdots_{x,y} & \cdots_{x,y} & \cdots_{x,y} \\ 0_{0,33} & 117_{1,33} & 117_{2,33} & 117_{3,33} & 0_{6,33} \\ 0_{0,34} & 117_{1,34} & 117_{2,34} & 117_{3,34} & 0_{6,34} \\ 0_{0,33} & 110_{1,35} & 110_{2,35} & 110_{3,35} & 0_{6,35} \\ 0_{0,34} & 110_{1,36} & 110_{2,36} & 110_{3,36} & 0_{6,36} \\ \cdots_{x,y} & \cdots_{x,y} & \cdots_{x,y} & \cdots_{x,y} & \cdots_{x,y} \end{bmatrix}$$

Figure 2.10 Matrice de sous-sections

Algorithme 2.3 Algorithme de calcul du centroïde

Input : 25 *section group, identified by a group identifier between* {***0, ... , 255***}
Output : 25 *centroids*
Centroids C_n
for *all section group* $S_{grp} \in \mathcal{S}_{grp}$ *(*$I_g = 1, ..., 25$*)* **do**
    Pixel positions $P_{pos} \leftarrow S_{grp}$ indexes x, y
    $X_{sum} \leftarrow \sum P_{pos} x$
    $Y_{sum} \leftarrow \sum P_{pos} y$
    $X_{mean} \leftarrow \frac{X_{sum}}{P_{pos} x_{count}}$
    $Y_{mean} \leftarrow \frac{Y_{sum}}{P_{pos} y_{count}}$
    $C_n \leftarrow (X_{mean}, Y_{mean})$
end

La classification des transformateurs est plus simpliste. Deux catégories ont été crées, “Bon transformateur” et “Mauvais transformateur”. Un transformateur avec une fuite d’huile est considéré comme étant un mauvais transformateur. Un transformateur sans fuite d’huile est quant à lui considéré comme étant bon. Ce système de classification ne gère que les scénarios de fuites d’huiles, puisque celui-ci est le scénario observable le plus commun pour un transformateur défectueux. Le système de classe a été intégré à Mask-RCNN en séparant la classe des transformateurs en deux classes, chaque classe représentant l’une des catégories.

2.3 Application Android

Pour regrouper et rendre accessible les fonctionnalités discutées précédemment, une application mobile Android a été mise sur pied. Cette application permet de recueillir des images en temps réel et d’analyser leur contenu afin d’établir des cotes d’états et de fournir des informations supplémentaires sur ce qui est observé.

Hydro-Québec possède une base de données contenant de l’information sur tous les équipements du réseau de distribution possédant une fiche signalétique LCLCL. Cette base de données contient plus de 1 millions d’enregistrements, soit une quantité trop élevée pour un appareil mobile. Ainsi, dans le but d’alléger et accélérer la recherche dans ces données, l’application télécharge localement une parcelle de la base de données, selon certains critères : tout équipement dans un rayon de 10 kilomètres selon la position actuelle de l’appareil mobile Android, rafraîchie si la nouvelle du sujet dépasse 5 kilomètres comparé à la position précédente ou si 15 minutes s’est écoulé depuis la dernière mise à jour.

Originalement, l’appareil Android exécutait les modèles de réseaux de neurones localement afin d’être quasi-indépendant d’un accès réseau, dans le but de supporter les régions plus éloignés du Québec. Dans la majorité des scénarios, les performances sont acceptables et l’appareil est stable. Toutefois, l’application devient instable et inutilisable si la température externe est trop élevée, due à des protections provenant du système de base de Android dans le but d’éviter des dommages matériels. Pour résoudre cette problématique, l’exécution des modèles a été transféré

vers un ordinateur portable. Ainsi, l'appareil mobile transmet par réseau sans-fil les images à l'ordinateur portable. Ensuite, l'ordinateur portable effectue la segmentation et obtient les cotes d'états au travers des réseaux de neurones et retourne à l'appareil mobile les résultats pour afficher l'information à l'utilisateur.

Lorsque l'appareil mobile détecte un poteau accompagné d'un transformateur et que dans un arc de 170° faisant face à l'utilisateur il existe un LCLCL correspondant, l'application affiche l'information attachée à ce LCLCL afin que l'utilisateur vérifie sa validité et puisse soumette le résultat de son inspection. Ce positionnement s'effectue à l'aide des multiples capteurs inclus avec les appareils mobiles modernes, principalement le capteur GPS, l'accéléromètre et le magnétomètre. En combinant les données des capteurs, il est possible de déterminer la position exacte de l'utilisateur et son orientation à l'aide de l'algorithme 2.4. La distance entre l'utilisateur et la plaque LCLCL est calculée à l'aide de la formule de haversine (Brummelen, 2013). L'orientation est quant à elle calculée à l'aide d'un algorithme utilisant la différence d'angle entre le coin supérieur gauche de l'appareil mobile, le pôle nord magnétique (azimut) et le LCLCL.

Algorithme 2.4 Algorithme de calcul de l'angle d'orientation

Input : *latitudeUser, longitudeUser, latitudeLCLCL, longitudeLCLCL, azimuth*
Output : *angle between both gps coordinates*
$Lat_u \leftarrow$ latitudeUser
$Lng_u \leftarrow$ longitudeUser
$Lat_l \leftarrow$ latitudeLCLCL
$Lng_l \leftarrow$ longitudeLCLCL
$Az \leftarrow$ azimuth
Convert the azimuth to clockwise angle
$Az = (Az + 360) \mod 360$
Counter-clockwise angle between the user and LCLCL. -90 to correct azimuth default angle.
$angle \leftarrow atan2(Lat_l - Lat_u, Lng_l - Lng_u) + 360 - 90) \mod 360$
Final angle, correcting to clockwise
$angle \leftarrow (-(angle + Az) + 360 * 2) \mod 360$

2.4 Segmentation à l'aide de réseaux de neurones adversaires

Avec une quantité de données étiquetées limitée, un volume important de données à traiter et un temps de travail restreint, une hypothèse a été émise : "Serait-il possible d'utiliser un réseau de type GAN afin d'effectuer l'étiquetage des images de façon automatique ?". Les approches de type GAN comportent deux réseaux en compétition : un premier réseau, le générateur, qui produit des données et un deuxième réseau, le discriminant, qui distingue si les données qui lui sont fournies sont des données réelles ou générées. Ici, le générateur serait un réseau de segmentation spécialement entraîné pour faire la segmentation des transformateurs et le discriminant, un réseau entraîné pour distinguer un vrai masque d'un faux.

Deux réseaux de segmentation sont intégrés au modèle GAN en tant que réseaux générateurs : un réseau simpliste pour la conception et l'architecture du réseau, composé d'une séquence variable de blocs de couches de convolutions, *batch normalization* et *leaky ReLU*. Une séquence miroir de blocs existe pour restaurer l'image à la taille originale, en remplacent la couche de convolution par une déconvolution. Le nombre de blocs diffère selon la taille des images utilisées. Pour l'architecture finale le deuxième réseau, Fast-SCNN (Poudel, Liwicki & Cipolla, 2019) a été sélectionné comme réseau générateur de masques dû à ses performances de pointe et sa compatibilité avec l'architecture générale du réseau GAN.

Le réseau discriminant est constitué d'une série de convolutions, *leaky ReLU* et *dropout*, finalisé par une couche de densification pour fusionner la valeur de sortie entre 0 et 1.

Au travers des itérations, ces deux réseaux dansent, cherchant à avoir le dessus sur l'autre réseau dans une compétition continue. Cette compétition arrive à terme lorsque le réseau de segmentation réussi à tromper, avec certitude, le réseau discriminant. Dans le but d'aider le réseau de segmentation, quelques tactiques sont utilisées. La première est un échantillon de données étiquetées mélangées au lot de données complet. Ces données permettent l'utilisation d'une fonction de coût par classes par pixel pour informer le réseau de segmentation de la qualité de ses prédictions, contrairement à la fonction du réseau discriminant qui ne donne qu'une valeur continue entre 0 et 1. La deuxième tactique est l'utilisation d'un réseau de détection d'objet, plus

facile à être entraîné avec peu de données, pour extraire l'objet à segmenter en pré-traitement. Cette étape facilite la tâche du réseau de segmentation en réduisant la portion de l'image à seulement l'objet qui doit être segmenté.

CHAPITRE 3

PRÉSENTATION DES RÉSULTATS

3.1 LCLCL synthétiques

Les résultats de la recherche sur les plaques LCLCL portent principalement sur le taux de succès de lecture des séquences de cinq caractères alphanumériques. Les modèles sont entièrement entraînés sur des données synthétiques et, par la suite, testés sur un sous-ensemble de test synthétique et 128 images réelles. L'hypothèse posée est qu'avec un volume assez élevé et varié de données synthétiques, il est possible d'entraîner un modèle de lecture de texte afin qu'il puisse prédire correctement dans plus de 80% (limite arbitraire) des cas les séquences de caractères.

Tableau 3.1 Performance des réseaux de lecture de plaques LCLCL synthétiques

	Réseau	**Type d'image**	**Taille**	**Succès (5/5) %**	**Succès (4+5/5) %**
1	DenseNet201	5xRGB	75x75	98.97	100
2	DenseNet201	5xRGB	75x150	98.42	100
3	DenseNet201	Gris	75x75	98.77	100
4	DenseNet201	Gris	150x150	99.22	100
5	GRU-Net	Gris	150x300	97.30	100
6	GRU-Net	Gris	75x75	98.23	100

Tableau 3.2 Performance des réseaux de lecture de plaques LCLCL réelles

	Réseau	**Type d'image**	**Taille**	**Succès (5/5) %**	**Succès (4+5/5) %**
1	DenseNet201	5xRGB	75x75	67.19	74.21
2	DenseNet201	5xRGB	75x150	65.63	80.47
3	DenseNet201	Gris	75x75	63.28	78.9
4	DenseNet201	Gris	150x150	60.94	78.13
5	GRU-Net	Gris	150x300	45.31	57.81
6	GRU-Net	Gris	75x75	40.63	61.72

Les résultats, résumés dans les tableaux 3.1, 3.2 et 3.3, montrent que les modèles n'arrivent pas à prédire dans plus de 80% des cas les séquences de caractères correctement. Par contre, en

Tableau 3.3 Performance des réseaux de lecture de plaques LCLCL. Séquences complètes

	Réseau	**5/5**	**4/5**	**3/5**	**2/5**	**1/5**	**0/5**
1	DenseNet201	86	9	9	10	6	8
2	DenseNet201	84	19	9	2	6	8
3	DenseNet201	81	20	5	6	4	12
4	DenseNet201	78	22	7	11	6	4
5	GRU-Net	58	16	8	11	13	22
6	GRU-Net	52	27	15	13	10	11

combinant les résultats de 5 caractères sur 5 et 4 caractères sur 5, plusieurs modèles sont très près ou franchissent l'objectif.

3.2 Segmentation des poteaux et transformateurs

La recherche sur les poteaux et transformateurs porte sur trois tâches : détecter, classifier et segmenter. Mask-RCNN, un réseau de neurones pour la segmentation par instance, est entraîné avec un lot de données composé de 300 images réelles étiquetées, provenant du jeu de données complet de 500,000 images, sélectionnées à la pièce pour avoir plus de diversité dans les images et des scénarios représentatifs du cas d'utilisation final, soit un technicien avec un appareil mobile. L'hypothèse va comme suit : est-ce qu'un seul réseau de neurones, sans pré-traitement ou post-traitement, est capable d'effectuer les trois tâches ?

Tableau 3.4 Métriques de Mask-RCNN

Perte	**Précision**	**Rappel**	**mAP**
0.8914	83.4%	76.1%	78.7%

Tel que présenté dans le tableau 3.4 et la figure 3.1, Mask-RCNN réussit à effectuer les trois tâches malgré la faible quantité de données utilisées pour l'entraînement. Il est donc confirmé qu'un seul réseau de neurones de segmentation par instance est capable de répondre à ces trois besoins.

Figure 3.1 Comparatifs de détection / segmentation avec la vérité

3.3 Application Android

L'application Android agit comme berceau pour les résultats de la recherche en plus d'offrir des fonctionnalités natives à elle-même. L'application Android étant interactive et réactive à son environnement à proximité, il devient donc difficile de trouver une métrique de mesure représentative de son fonctionnement. Étant au centre de la recherche, l'application Android est

donc évaluée selon un critère d'unification des fonctionnalités, soit l'hypothèse qu'il est possible d'effectuer de la reconnaissance, segmentation et classification en temps réel en plus d'établir des mesures de cotes d'état.

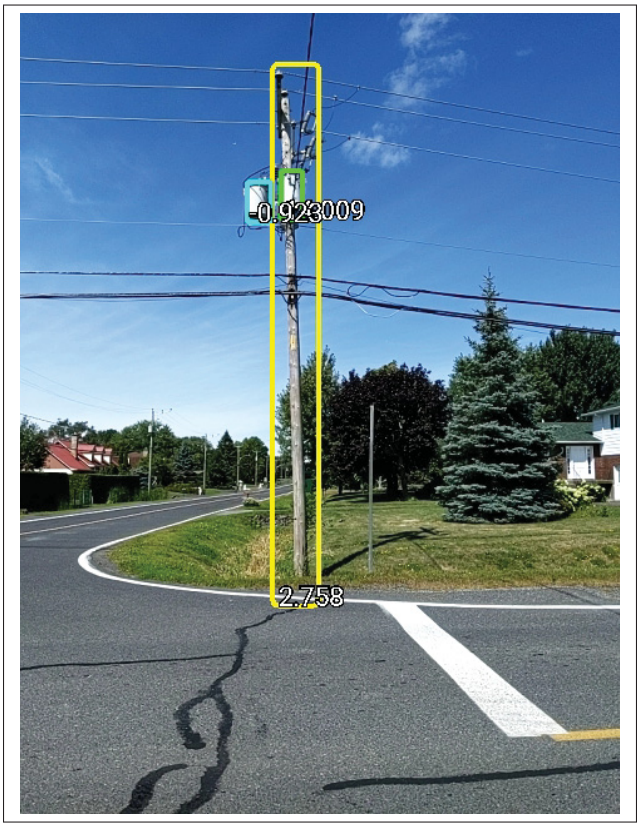

Figure 3.2 Réalité augmentée sur l'application Android

La figure 3.2 montre le résultat d'une analyse temps-réel effectuée par l'application. Cette analyse identifie les poteaux et transformateurs dans l'image et calcule leurs angles respectifs, comparé au plan absolu provenant de l'accéléromètre de l'appareil mobile.

La liste des équipements à proximité, montré à la figure 3.3, permet à l'utilisateur de s'orienter vers les équipements qu'il désire analyser. Les flèches s'orientent selon la position et l'orientation de l'appareil mobile. La distance est calculée à l'aide de la position GPS de l'appareil mobile et l'équipement, selon une ligne droite directe.

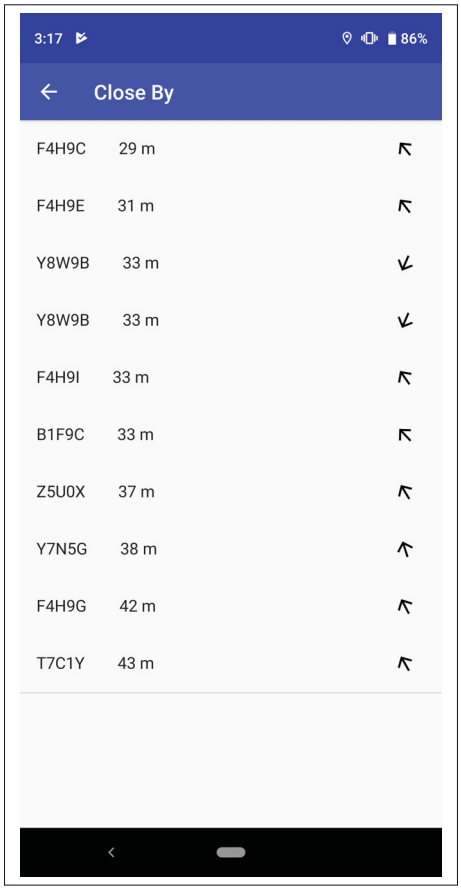

Figure 3.3 Liste indicatrice des équipements à proximité

Tous les résultats de la recherche étant implantés dans l'application Android, celle-ci répond donc aux besoins et, dans un même temps, répond de manière satisfaisante à l'hypothèse.

3.4 Segmentation par réseaux de neurones adversaires

L'hypothèse de la recherche sur les réseaux de neurones adversaires est la suivante : est-il possible d'effectuer de la segmentation sémantique en utilisant un grand volume de données non-étiquetées en combinant un réseau de segmentation et un réseau discriminant ?

Les figures 3.4, 3.5 et 3.6 montrent l'évolution progressive de l'apprentissage du réseau adversaire. À gauche, le résultat de l'inférence du réseau, au centre la vérité et à droite l'image originale.

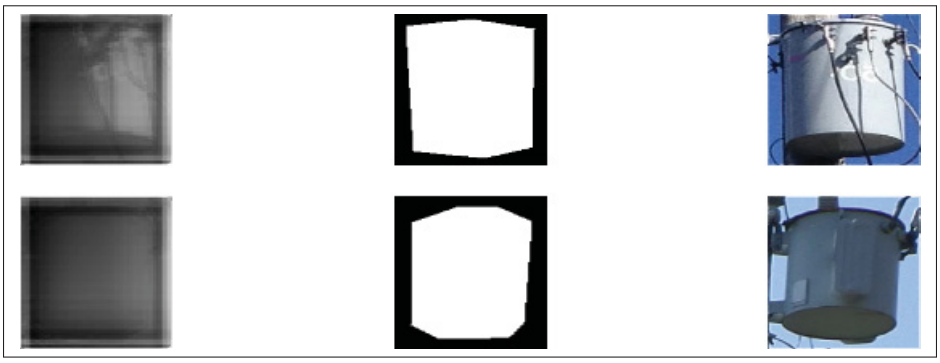

Figure 3.4 Apprentissage adversaire sur les transformateurs. 6ème époque

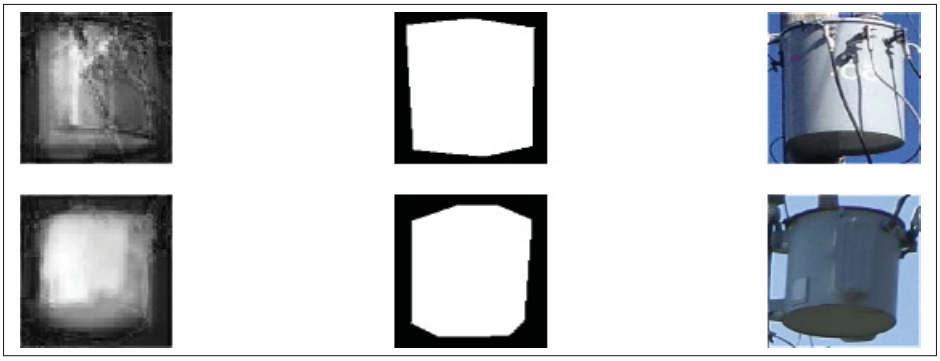

Figure 3.5 Apprentissage adversaire sur les transformateurs. 10ème époque

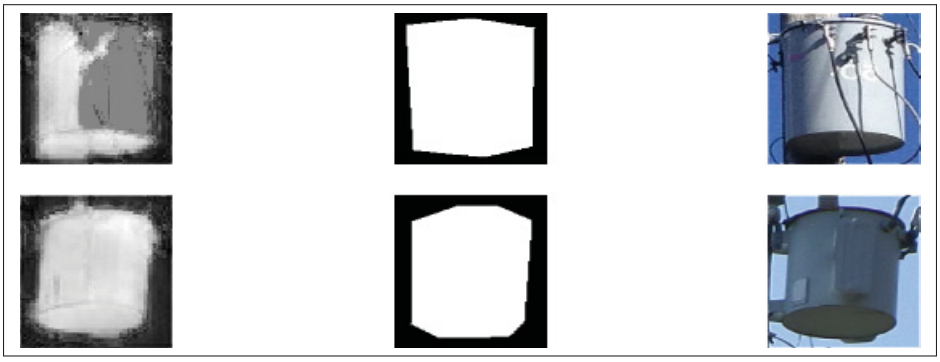

Figure 3.6 Apprentissage adversaire sur les transformateurs. 16ème époque

Les résultats montrent que le réseau de segmentation sémantique par apprentissage adversaire arrive progressivement à créer une segmentation se rapprochant de la vérité recherchée. Par contre, cette méthode ne fonctionne que dans les scénarios avec peu de bruit dans l'image et avec un zoom sur l'objet dont la segmentation est désirée. Avec des résultats imparfaits et une architecture compatible seulement avec certains types d'images, les résultats sont considérés inconcluants.

CHAPITRE 4

INTERPRÉTATION DES RÉSULTATS

Certains résultats obtenus répondent aux attentes, alors que d'autres n'atteignent pas les objectifs visés. Néanmoins, les résultats les plus intéressants démontrent la valeur de poursuivre la recherche dans ce domaine. Afin d'identifier ces résultats prometteurs, une analyse subjective des résultats a été effectuée. Cette analyse interprète les résultats obtenus et identifie les portions où il y a eu de bonnes manipulations, des erreurs dans les interprétations et techniques et finalement, révise la qualité des calculs faits.

4.1 LCLCL synthétiques

La problématique LCLCL peut être séparée en deux : le manque de données (étiquetées ou non) et la lecture du texte vertical. Globalement, les résultats sont impressionnants considérant la première approche utilisée afin de générer les LCLCL synthétiques, un algorithme qui reflète un modèle d'une plaque LCLCL dans divers scénarios. C'est la combinaison de plusieurs approches (variation des tons de couleurs, faux-dommages, faux-reflets) et algorithmes (*skew*, *projection*, binarisation) qui a permis d'obtenir des LCLCL synthétiques presque indiscernables par un humain lorsque comparés à des plaques réelles. DenseNet et GRU-Net, entraînés avec les données synthétiques, ont parfois des difficultés à bien reconnaître certains caractères sous certaines conditions, la plus évidente étant la source de l'image (photo, capture d'écran, image synthétique). Malgré la similitude visuelle entre les plaques synthétiques et réelles, les données numériques sous-adjacentes diffèrent et donc les modèles induisent des prédictions erronées. Ainsi, une similitude visuelle n'est pas suffisante pour entraîner un réseau de neurones sur un jeu de données synthétiques. L'utilisation de techniques d'augmentation d'images, tel qu'un filtre gaussien (voir la figure 4.1), pourrait possiblement améliorer la qualité des modèles entraînés avec les dites images augmentées. Le filtre gaussien permet aussi de réduire l'impact du bruit et des défauts mineurs dans les images.

Figure 4.1 Gauche : Image sans filtre gaussien.
Droite : Image avec filtre gaussien

Les données montrent qu'un seul caractère vertical de la plaque LCLCL réelle sur cinq est plus fréquemment mal reconnu, comparé aux autres caractères et prédictions effectuées sur des données synthétiques. Cette anomalie provient d'une erreur dans l'utilisation de l'algorithme de *skew*. *Skew* requiert un point d'ancrage pour effectuer la déformation de l'image. Celui-ci a été incorrectement localisé dans le haut de l'image, causant une distorsion progressive autour de celui-ci lorsque la déformation est appliquée (voir figure 4.2). Un correctif serait de localiser le point d'ancrage dans le haut de l'image, mais à l'extérieur de celle-ci.

Malgré ces défauts, l'utilisation astucieuse de techniques de manipulation et combinaison d'images, telles que celle utilisé par le FSNS Dataset (Smith *et al.*, 2016), où quatre images de perspectives différentes du même sujet sont combinées afin de faire une prédiction unique, ont permis de faire des bons majeurs dans la qualité de détection du modèle de DenseNet. Ici, en combinant cinq images synthétiques augmentées pour améliorer l'efficacité (Mikołajczyk & Grochowski, 2018), les performances du réseau ont atteint et même dépassé celles de GRU-Net.

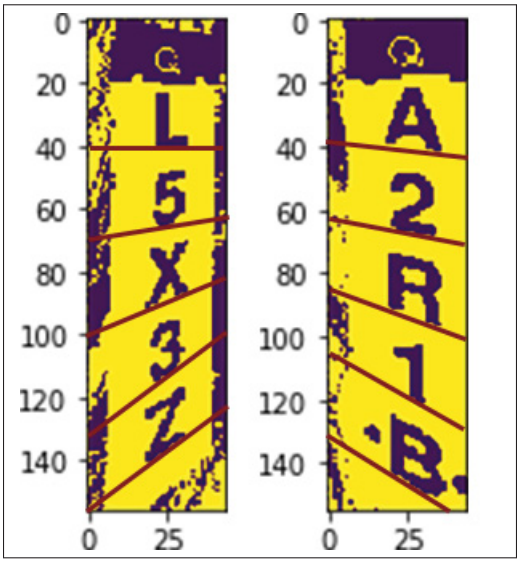

Figure 4.2 Distorsion progressive dans les LCLCL synthétiques

Par contre, avec seulement quatre caractères reconnus, il est tout de même possible d'inférer quelle est la bonne plaque LCLCL grâce à la combinaison de la séquence de caractères et les données des équipements géo-référencés. Ces deux données permettent de calculer la proximité entre les séquences et leurs similitudes, permettant ici de déterminer quelle est la bonne séquence LCLCL.

4.2 Segmentation et classification des poteaux et transformateurs

Le modèle utilisé pour la segmentation des poteaux et transformateurs, Mask R-CNN, de par son architecture, donne de bons résultats malgré le faible volume initial de données étiquetées. Les 230 images utilisées en entraînement et 70 réservées pour la validation, contenant toutes au minimum un poteau ayant de zéro à trois transformateurs attachés, sont assez variées pour entraîner un modèle de segmentation capable de segmenter la majorité des poteaux composants le réseau de distribution d'électricité d'Hydro-Québec. Il y a quelques exceptions où le modèle est incapable de détecter et segmenter les poteaux. La plus importante et fréquente étant les poteaux ayant reçu un traitement chimique. Ce traitement chimique donne à ces poteaux un ton

verdâtre, un ton inexistant dans le jeu de données utilisé pour l'entraînement. Effectivement, depuis l'an 2000, les nouveaux poteaux installés sont traités chimiquement pour prévenir les dommages infligés par les insectes et certains autres types de dommages causés par la nature. Entre autres, la banque d'un demi-million d'images ne possède que des images de poteaux ayant subis des dommages et possédant des défauts. Par conséquent, la majorité de ces poteaux est en fin de vie et donc, précède le début du traitement chimique.

Dans le but d'obtenir un modèle plus versatile et robuste, sans avoir immédiatement recours à l'ajout de données supplémentaires, quelques techniques simples de manipulation d'image pourraient être mises à l'essai. En pré-traitement, travailler en tons de gris et éliminer les couleurs pourraient améliorer le taux de détection des poteaux mais au risque de réduire la qualité des détections et segmentations. Une autre approche serait d'unir les espaces de couleurs communs aux poteaux et remplacer ces couleurs par une couleur unique. Cette solution résout la problématique du changement de couleur dû au traitement chimique mais rend le modèle dépendant du pré-traitement et sa qualité de détection/segmentation en découle directement. La solution la plus appropriée à la problématique des nouveaux poteaux serait de prendre une caméra, un véhicule et faire une tournée du Québec à la recherche de poteaux ayant subit un traitement chimique.

4.3 Application Android

L'application Android répond aux besoins exprimés par l'IREQ et implémente les fonctionnalités développées au cours du projet de recherche. Son architecture, originalement basée sur des services et le patron de programmation *publish-subscribe*, n'a pas été respectée durant l'avancement du projet. Cette divergence est principalement attribuée à l'inexpérience avec les outils et l'API de développement d'Android. L'apprentissage du fonctionnement des méthodes de cet API et interactions avec les senseurs du téléphone se sont fait en même temps que les fonctionnalités étaient implémentées.

Une ré-écriture complète de l'application serait préférable, maintenant que la liste des fonctionnalités complète est connue et les particularités de l'API d'Android ont été maîtrisées.

4.4 Segmentation à l'aide de réseaux de neurones adversaires

L'expérience avec les GANs montre des résultats inspirants, mais incomplets. L'objectif du travail avec les GANs est d'automatiser la segmentation des poteaux et transformateurs afin de pouvoir utiliser les 500,000 images de l'ensemble de données d'Hydro-Québec. Les résultats sont inspirants puisque certaines expérimentations, dans des conditions simplistes, donnent de bons résultats (voir par exemple la figure 4.3). Dans les cas complexes, par exemple segmenter un objet parmi un décor, les résultats deviennent inutilisables.

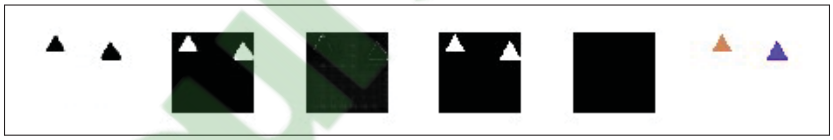

Figure 4.3 De gauche à droite : Arrière-plan, triangles segmentés, rectangles segmentés, vérité triangles, vérité rectangles, image originale

Avec des données simples, telles que des formes géométriques (voir figure 4.3), le réseau réussi à segmenter correctement des données sans étiquettes, ayant seulement l'opinion du réseau discriminant comme source d'information pour s'améliorer. Ce succès est attribuable à un facteur : la faible complexité des images produites par l'algorithme. Les images sont simples, les formes utilisent des couleurs unies, sont distinctives et sont positionnées sur un fond blanc. Il y a aussi une absence complète de bruit. Lorsque des images réelles sont utilisées, la qualité des segmentations chute drastiquement.

Ces mauvais résultats peuvent être expliqués par un manque de connaissances envers le fonctionnement des réseaux de neurones de type GAN. Effectivement, un élément crucial du générateur est que les données en entrée doivent respecter une loi normale ou une distribution uniforme (Goodfellow *et al.*, 2014) pour générer un résultat cohérent. Dans le cas d'utilisation

souhaité, les données utilisées en entrée, soient les images, sont statistiquement aléatoires. Les cas simples fonctionnent dû au nombre limité de paramètres, l'absence de bruit en arrière plan et la simplicité des formes à segmenter.

Un autre élément qui impacte les résultats est l'utilisation d'une version beta (2.0.0-beta0) de TensorFlow 2. Cette version de TensorFlow possède des défauts qui limite l'entraînement des réseaux de neurones de type GAN, tel qu'une fuite de mémoire majeure qui cause l'entraînement à être arrêté prématurément et un problème lors du calcul du gradient, qui cause celui-ci à exploser. Ainsi, il est impossible d'effectuer un entraînement sur une longue période de temps.

CHAPITRE 5

DISCUSSION DES RÉSULTATS

5.1 Méthodes de génération de LCLCL synthétiques et raffinements

L'idée derrière l'utilisation de plaques LCLCL synthétiques est de permettre d'entraîner un modèle de lecture de texte vertical avec une quantité quasi-illimitée d'exemples de textes représentatifs des plaques LCLCL réelles, puisque la quantité de données annotées de ces dernières est très limitée et que d'effectuer un inventaire poussé des plaques réelles est très coûteux.

La synthétisation des plaques LCLCL fait face à une problématique connue dans le domaine de la ressemblance humaine, la vallée dérangeante (*uncanny valley* en anglais) (Kätsyri, Mäkäräinen & Takala, 2017). La figure 5.1 montre que, jusqu'à un certain point, la vraisemblance augmente progressivement et une fois qu'elle atteint un certain seuil, il y a une chute brusque dans la vraisemblance perçue pour finalement, remonter et atteindre l'objectif visé. Cette vallée, ce gouffre, existe aussi pour les plaques synthétiques LCLCL. Ces dernières progressent dans leur vraisemblance en ajoutant de plus en plus de détails, en jouant sur la perspective, recrée les effets de l'ensoleillement, etc. Cette augmentation de la quantité d'information repose sur une balance très fragile : la quantité d'information fournie par les plaques LCLCL synthétique est adéquate, représentative des plaques LCLCL réelles et le modèle ne démontre pas de différences entre les prédictions sur les plaques réelles ou synthétiques ou le modèle est incapable de faire une prédiction parfaite sur les plaques réelles.

Avec un problème d'équilibre précaire, une quantité limitée de plaques réelles et quasi-illimitée de synthétiques, une approche basée sur (Inoue *et al.*, 2018) a été mise à l'essai. L'idée ici est de synthétiser des plaques réelles en utilisant un auto-encodeur entraîné à encoder des images de plaques LCLCL réelles et décoder des images de plaques LCLCL synthétiques (voir figure 5.2). En renversant ainsi le problème, la vallée dérangeante peut être évitée et le besoin de générer des plaques réalistes est dramatiquement réduit. Ce procédé est mis de

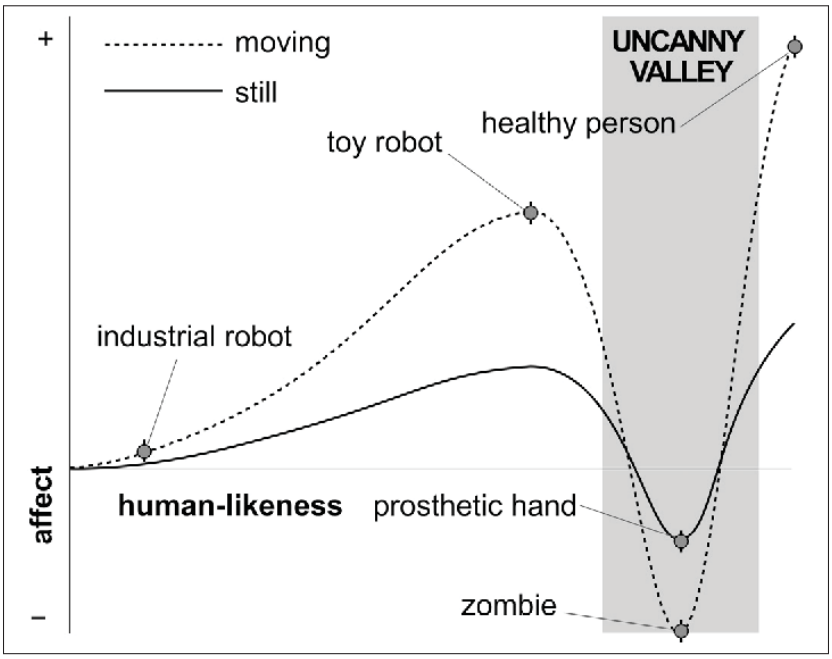

Figure 5.1 Vallée dérangeante (Uncanny valley).
Tirée de Kätsyri *et al.* (2017, p. 150)

l'avant comme étant une solution lorsque la quantité de données est limitée, cependant souffre d'une problématique qui s'avère fatale au cas d'utilisation avec les plaques LCLCL. Cette problématique tourne autour de la simplicité du sujet à synthétiser. Dans le cas d'utilisation mis de l'avant dans la publication, il s'agit d'un cylindre sur une surface plane. Outre la position de l'objet sur le plan, il y a très peu de variations. Une façon imagée de comprendre le fonctionnement de l'auto-encodeur est que celui-ci mémorise la position du cylindre et repositionne son équivalent synthétique sur le plan. Dans le cas des LCLCL, l'auto-encodeur mémorises certaines lettres/chiffres du Lettre-Chiffre-Lettre-Chiffre-Lettre et génère des LCLCL synthétiques, tous identiques. Une cause probable de cette problématique est un manque de données réelles pour l'entraînement de l'auto-encodeur puisque le domaine de variabilité des plaques LCLCL est énorme : $26 \times 10 \times 26 \times 10 \times 26 = 1,757,600$ combinaisons possibles.

L'approche de génération par algorithme étant trop complexe et la synthétisation des images réelles presque impossible dû à la grandeur du domaine, la solution de raffinement par GAN, selon (Shrivastava *et al.*, 2017), a donc été étudiée. Cette solution utilise des données existantes

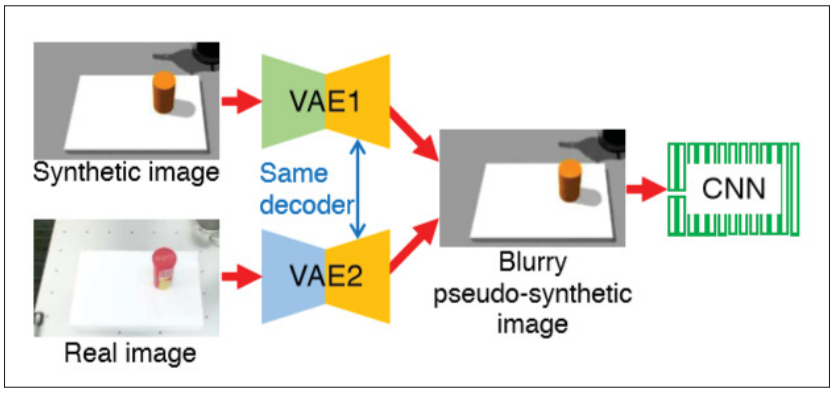

Figure 5.2 Architecture de synthétisation des images. Tirée de Inoue *et al.* (2018, p. 1)

synthétiques et à l'aide d'un réseau GAN et de données réelles comme repères, transforme les données synthétiques en données réelles. Ce raffinement referme l'écart si difficile à traverser entre les données synthétiques et réelles en minimisant la différence par pixel, donnant des valeurs plus réalistes et compatibles avec ce qu'il serait possible de retrouver dans le monde réel.

La génération et le positionnement des caractères sur la plaque LCLCL est imparfait, tel que démontré par la distorsion dans l'alignement de ceux-ci. Il est possible que l'étape de raffinement corrige ce défaut, mais il est préférable de régler le problème lors de la génération des plaques synthétiques. Une solution qui mérite d'être mise à l'essai est l'utilisation d'une carte de profondeur, tel que proposé par (Gupta *et al.*, 2016) lors du positionnement du texte synthétique dans les images réelles dans la recherche. Cette carte de profondeur peut être générée conjointement avec l'étape de *skew* de l'algorithme de génération des images synthétiques.

5.2 Segmentation avec données partiellement étiquetées de poteaux et transformateurs

La segmentation par réseaux adversaires est un domaine pour lequel il n'y a pas encore beaucoup de recherche, mais dont les possibilités sont immenses. Les travaux de recherche de (Luc *et al.*, 2016 ; Zhang *et al.*, 2017) agissent comme fondation pour l'architecture du réseau adversaire utilisée pour segmenter les poteaux et transformateurs. Effectivement, le modèle hybride à double perte permet au réseau d'apprendre beaucoup plus rapidement et précisément en utilisant

un échantillon de données annotées pour diriger l'apprentissage. Cette approche permet d'obtenir des résultats satisfaisants sur des données simples et sans bruit (voir figures 4.3 et 3.6). Par contre, lorsque les objets à segmenter sont petits ou s'il y a beaucoup d'information dans l'image (voir figure 5.3), le réseau est incapable de trouver l'objet et le segmenter adéquatement. Une explication possible est que le réseau de segmentation choisi, FastSCNN, n'est pas optimisé pour effectuer la segmentation de petits objets. Il existe des solutions spécialisées, tel que *Objects as Points* (Zhou, Wang & Krähenbühl, 2019), pour identifier les petits objets. *Objects as Points* modélise l'objet en un seul point et régresse le contour de l'objet à partir de ce point. Peut-être qu'en combinant les approches, il serait possible de gérer ces scénarios plus difficiles.

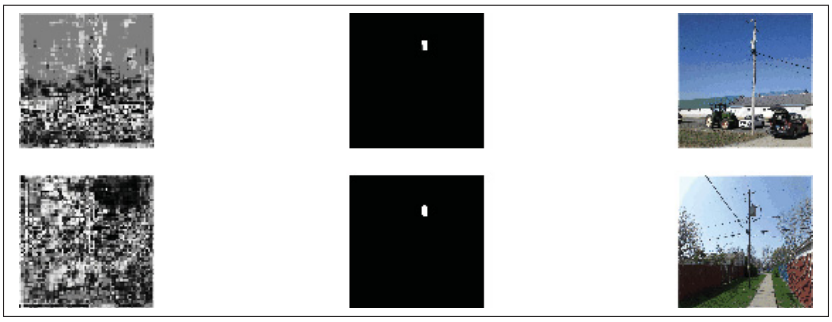

Figure 5.3 Mauvais résultats de la segmentation par GAN

Utiliser le co-entraînement (Zhang *et al.*, 2018) entre une architecture de détection de petits objets pour extraire ceux-ci et un réseau de segmentation sémantique pour segmenter les objets extraits est une autre solution potentiellement intéressante. L'utilisation d'un troisième réseau dans une architecture adversaire coopérative (voir figure 5.4) est un sujet de recherche qui commence à faire une percée dans le domaine. Ici, le réseau de détection et segmentation utilisent le même discriminant permettant le partage des erreurs et l'amélioration de leur voisin respectif. De plus, le réseau de segmentation alimente en données supplémentaires le réseau de détection puisque, si la segmentation est bonne, la détection l'est aussi.

La fonction de coût du réseau discriminant de (Hung *et al.*, 2018) mesure pour chaque portion de l'espace vectoriel de l'image si la prédiction sur le type de segmentation (manuelle, automatique) est correcte. Cette approche pourrait remplacer la solution actuelle où le discriminant fait sa

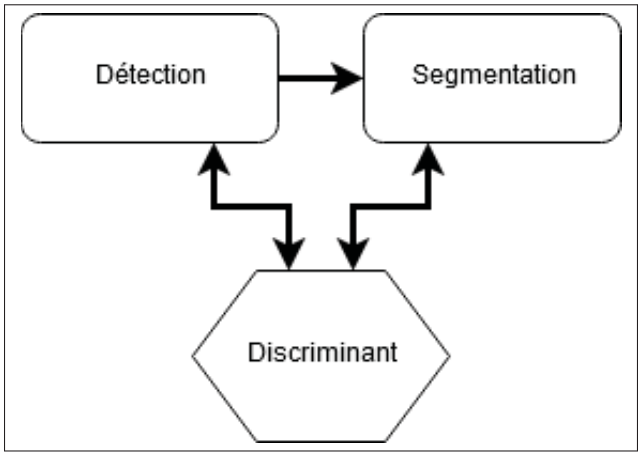

Figure 5.4 Réseau adversaire coopératif

prédiction sur les classes de segmentation, une prédiction sans le contexte de l'image originale, afin d'enrichir la quantité de données à la disposition du discriminant pour effectuer sa prédiction.

CONCLUSION ET RECOMMANDATIONS

La pérennisation du réseau électrique est devenu un sujet vif au Québec, suite à une panne majeure en novembre 2019 entraînant plus d'un million d'usagers dans le noir. Cette panne a fait surgir aux yeux du public l'importance de la maintenance et le remplacement des équipements pour le transport et la distribution d'électricité au Québec. L'apport d'outils spécialisés tel que proposé par la recherche pourrait autant accélérer la remise en état fonctionnel du service que de permettre de prévenir de tels scénarios, en faisant une maintenance préventive sur les points les plus critiques du réseau électrique, les équipements et actifs en défauts.

Ces outils permettent de repérer, identifier et analyser en temps-réel l'état des poteaux et transformateurs faisant parti du réseau de distribution qui s'étale sur plusieurs millions de kilomètres. Ce sont certaines percées dans le domaine de l'apprentissage-machine et les réseaux de neurones profonds qui permettent à ce type d'outils d'apparaître sur le marché de nos jours. La réalisation du réseau de segmentation par instance Mask-RCNN a permis à la recherche de réaliser une architecture de prédictions précises avec un faible volume de données étiquetées. L'intérêt du domaine envers les réseaux adversaires va ouvrir, d'ici quelques années, les portes vers une multitude de nouvelles solutions de tous genres, telle qu'un entraînement de qualité avec des données faiblement étiquetées.

C'est ici que la recherche se conclut, mais la solution n'est qu'à son début de vie. Au final, il s'agit d'un effort collectif mondial, grâce aux multiples publications publiques, qui va permettre au réseau électrique du Québec de retrouver un état fonctionnel de meilleur qualité.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Badrinarayanan, V., Kendall, A. & Cipolla, R. (2017). SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 2481-2495.

Bai,W., O. O. S. M. S. H. R. M. T. G. G. B. K. A. M. P. & Rueckert, D. (2017). Semi-supervised learning for network-based cardiac MR image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham*, 253-260.

Bilen, H. & Vedaldi, A. (2015). Weakly Supervised Deep Detection Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2846-2854.

Brummelen, G. V. (2013). *Heavenly Mathematics : The Forgotten Art of Spherical Trigonometry*. Princeton University Press.

Cho, K., van Merrienboer, B., Bahdanau, D. & Bengio, Y. (2014). On the Properties of Neural Machine Translation : Encoder-Decoder Approaches.

Chollet, F. (2016, Mai, 14). Building Autoencoders in Keras. Repéré à https://blog.keras.io/building-autoencoders-in-keras.html.

Cinbis, R. G., Verbeek, J. & Schmid, C. (2017). Weakly Supervised Object Localization with Multi-Fold Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 189-203.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative Adversarial Nets. 2672–2680. Repéré à http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Graves, A., Fernández, S. & Gomez, F. (2006). Connectionist temporal classification : Labelling unsegmented sequence data with recurrent neural networks. *In Proceedings of the International Conference on Machine Learning, ICML 2006*, pp. 369–376.

Gupta, A., Vedaldi, A. & Zisserman, A. (2016). Synthetic Data for Text Localisation in Natural Images. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2315-2324.

He, K., Gkioxari, G., Dollár, P. & Girshick, R. B. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980-2988.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261-2269.

Hui, J. (2018, Avril, 19). Image segmentation with Mask R-CNN. Repéré à https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272.

Hung, W.-C., Tsai, Y.-H., Liou, Y.-T., Lin, Y.-Y. & Yang, M.-H. (2018). Adversarial Learning for Semi-supervised Semantic Segmentation. *BMVC*.

Hydro-Québec. (2019, Novembre, 3). Hydro-Québec fait le point sur les pannes d'électricité [Communiqué de presse]. Repéré à http://nouvelles.hydroquebec.com/fr/communiques-de-presse/1560/hydro-quebec-fait-le-point-sur-les-pannes-delectricite/.

Inoue, T., Chaudhury, S., Magistris, G. D. & Dasgupta, S. (2018). Transfer Learning from Synthetic to Real Images Using Variational Autoencoders for Precise Position Detection. *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2725-2729.

Jaderberg, M., Simonyan, K., Zisserman, A. & kavukcuoglu, k. (2015). Spatial Transformer Networks. Dans Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R. (Éds.), *Advances in Neural Information Processing Systems 28* (pp. 2017–2025). Curran Associates, Inc. Repéré à http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf.

Kätsyri, J., Mäkäräinen, M. & Takala, T. (2017). Testing the 'uncanny valley' hypothesis in semirealistic computer-animated film characters : An empirical evaluation of natural film stimuli. *Int. J. Hum.-Comput. Stud.*, 97, 149-161.

Kingma, D. P. & Welling, M. (2013). Auto-Encoding Variational Bayes. *CoRR*, abs/1312.6114.

Kullback, S. & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86. Repéré à http://www.jstor.org/stable/2236703.

Liu,W., A. D. E. D. S. C. R. S. F. C. & Berg, A. (2016). SSD : Single shot multibox detector. *European conference on computer vision, Springer, Cham*, 21-37.

Luc, P., Couprie, C., Chintala, S. & Verbeek, J. (2016). Semantic Segmentation using Adversarial Networks. *ArXiv*, abs/1611.08408.

Mikołajczyk, A. & Grochowski, M. (2018, May). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 117-122. doi : 10.1109/IIPHDW.2018.8388338.

Nikolenko, S. I. (2019). Synthetic Data for Deep Learning.

Pérez, P., Gangnet, M. & Blake, A. (2003). Poisson Image Editing. *ACM Trans. Graph.*, 22(3), 313–318. doi : 10.1145/882262.882269.

Poudel, R. P. K., Liwicki, S. & Cipolla, R. (2019). Fast-SCNN : Fast Semantic Segmentation Network. *ArXiv*, abs/1902.04502.

Ren, S., He, K., Girshick, R. B. & Sun, J. (2015). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 1137-1149.

Shi, B., Bai, X. & Yao, C. (2017). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11), 2298-2304.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W. & Webb, R. (2017). Learning from Simulated and Unsupervised Images through Adversarial Training. 2242-2251.

Smith, R., Gu, C., Lee, D.-S., Hu, H., Unnikrishnan, R., Ibarz, J., Arnoud, S. & Lin, S. (2016). End-to-End Interpretation of the French Street Name Signs Dataset. *Computer Vision – ECCV 2016 Workshops*, pp. 411–426.

Souly, N., Spampinato, C. & Shah, M. (2017). Semi Supervised Semantic Segmentation Using Generative Adversarial Network. *2017 IEEE International Conference on Computer Vision (ICCV)*, 5689-5697.

Tang, Y., Wang, J., Gao, B., Dellandréa, E., Gaizauskas, R. J. & Chen, L. (2016). Large Scale Semi-Supervised Object Detection Using Visual and Semantic Knowledge Transfer. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2119-2128.

Ulyanov, D., Vedaldi, A. & Lempitsky, V. S. (2017). Deep Image Prior. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9446-9454.

Yan, Z., Liang, J., Pan, W., Li, J. & Zhang, C. (2017). Weakly- and Semi-Supervised Object Detection with Expectation-Maximization Algorithm. *ArXiv*, abs/1702.08740.

Zhang, Y., Xiang, T., Hospedales, T. M. & Lu, H. (2018, June). Deep Mutual Learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4320-4328.

Zhang, Y., Yang, L., Chen, J., Fredericksen, M., Hughes, D. P. & Chen, D. Z. (2017). Deep Adversarial Networks for Biomedical Image Segmentation Utilizing Unannotated Images. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, pp. 408-416.

Zhou, X., Wang, D. & Krähenbühl, P. (2019). Objects as Points.

Zhou, Y., Wang, Y., Tang, P., Bai, S., Shen, W., Fishman, E. K. & Yuille, A. L. (2018). Semi-Supervised Multi-Organ Segmentation via Deep Multi-Planar Co-Training.