

Ce poster est le résultat de mon stage de fin de troisième année à l'École Nationale des Sciences Géographiques. Ce stage a été réalisé du 1er Mai au 13 Septembre au département des sciences géographiques de l'université d'Ottawa. Le but de mon stage a été d'étudier l'accessibilité urbaine de la ville d'Ottawa à l'aide de la technologie deep-learning. L'accessibilité urbaine est un phénomène complexe qui mesure l'envie d'une personne de marcher dans telle ou telle rue. Sa mesure se base sur des éléments concrets : présence de trottoirs, sécurité du quartier, présence d'espaces verts, météo, richesse du quartier, proximité avec les commerces, musées, parcs etc... Mon objectif principal a été de donner une mesure de l'accessibilité urbaine de la ville d'Ottawa et de la comparer à d'autres résultats existants. Une fois cette étape complétée, j'ai appliqué les dernières avancées scientifiques en matière de vision par ordinateur par deep-learning au domaine de l'accessibilité urbaine d'Ottawa.

## 1) L'ensemble d'apprentissage :

N'importe quel réseau de neurones a besoin d'un ensemble de données pour apprendre à réaliser la tâche qui lui est demandée. Mon ensemble d'apprentissage est composé de 2257 duels parmi 3765 photos Google StreetView d'Ottawa. Chaque duel est constitué d'un affrontement entre 2 images sur un critère d'accessibilité urbaine. La figure 1 à droite montre des images extraites de l'ensemble d'apprentissage. On remarquera que les photos sont orientées selon un axe parallèle à celui de la route.



Figure 1 : Images extraites de l'ensemble d'apprentissage

## 2) Mesure de l'accessibilité urbaine :

### ScoreCroutiNET :

Comme le montre la figure ci-dessous, cette architecture prend une image en entrée et ressort le score associé. De plus, un réseau auxiliaire est utilisé pendant la phase d'apprentissage pour apprendre à partir de duels.

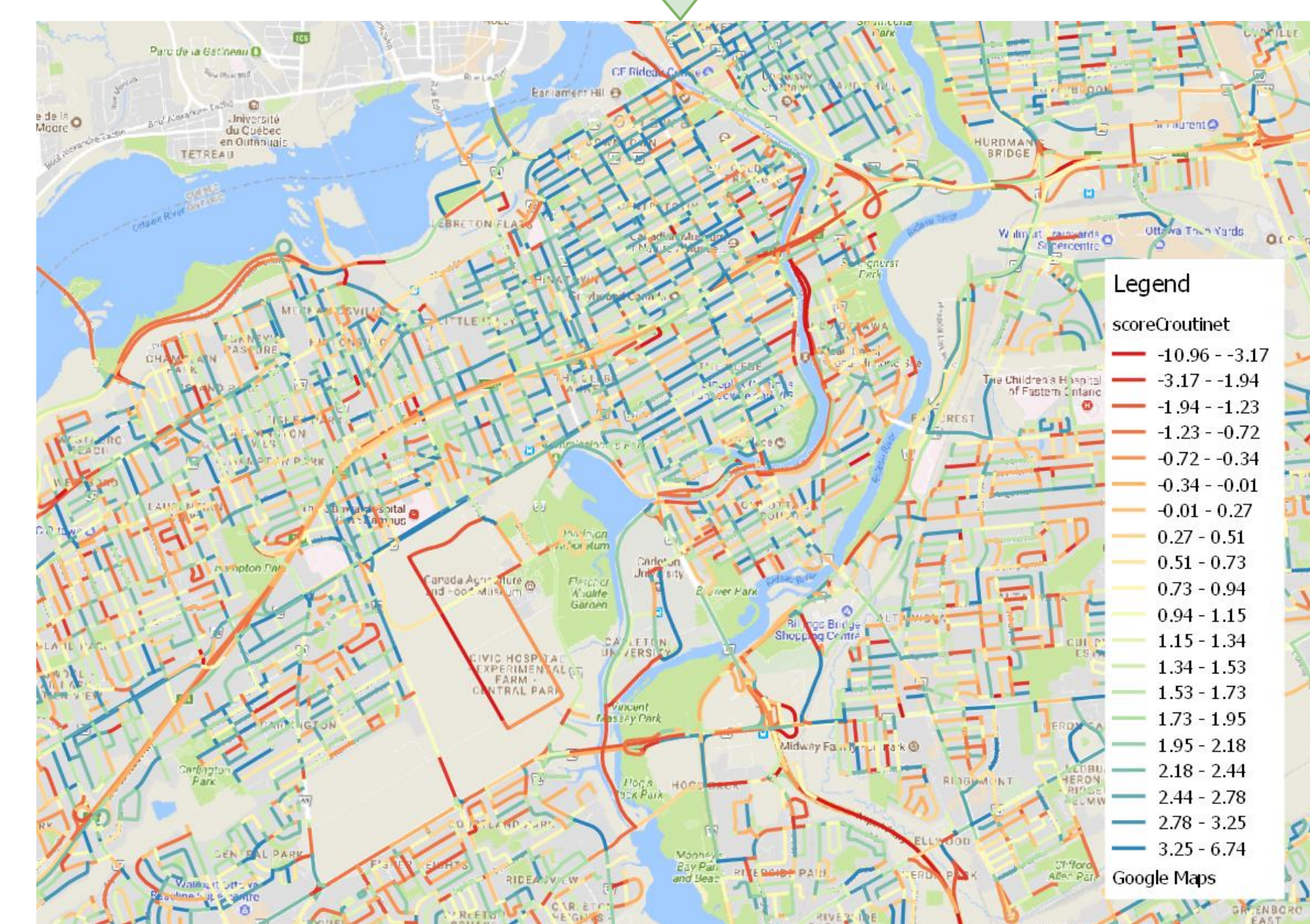
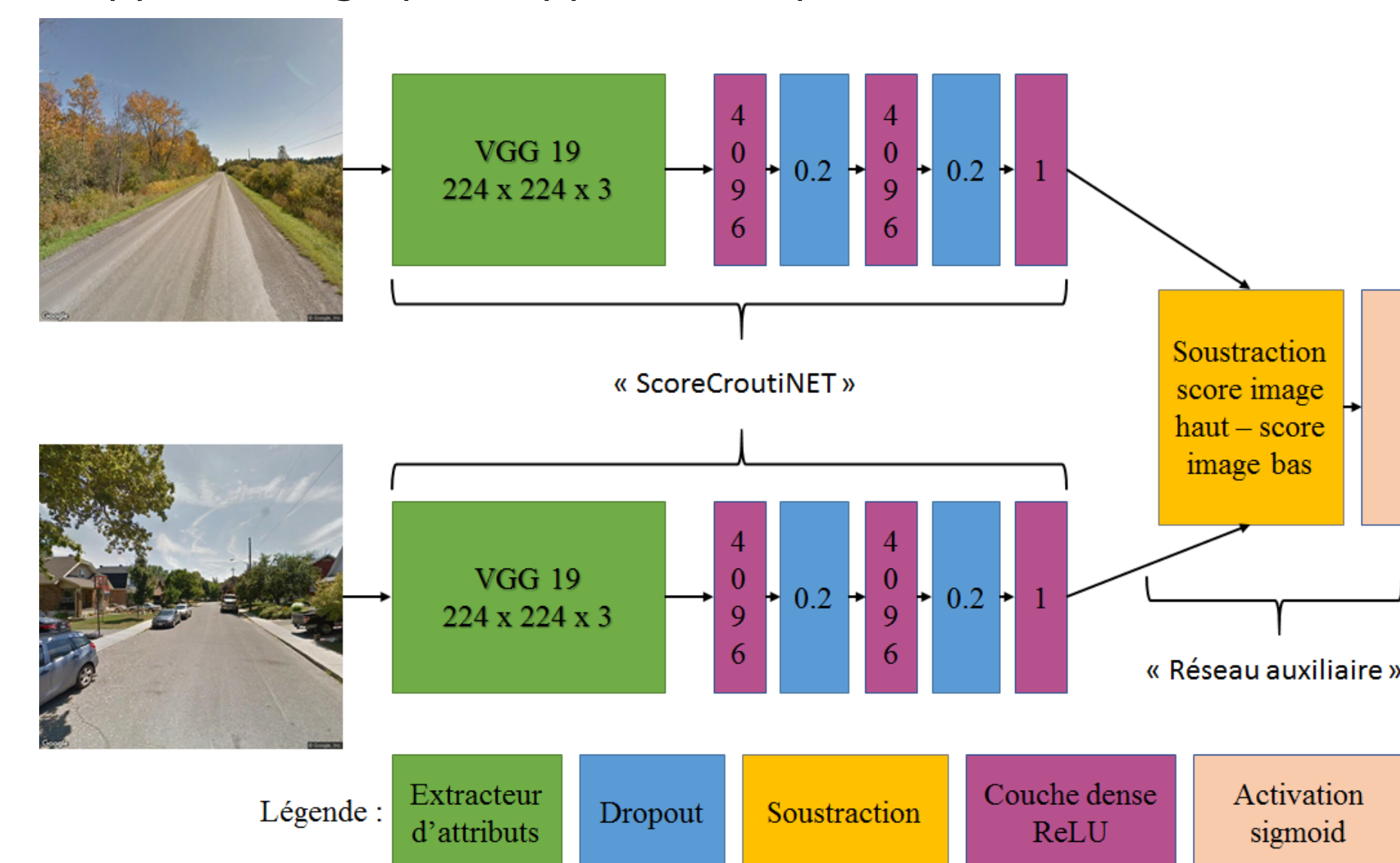


Figure 2.a : ScoreCroutiNET

### ComparaisonCroutiNET :

Basé sur l'architecture SS-CNN (DUBEY et al. 2016), ce réseau prend deux images en entrée et résout leur duel, selon un critère d'accessibilité urbaine. Son architecture est détaillée ci-dessous :

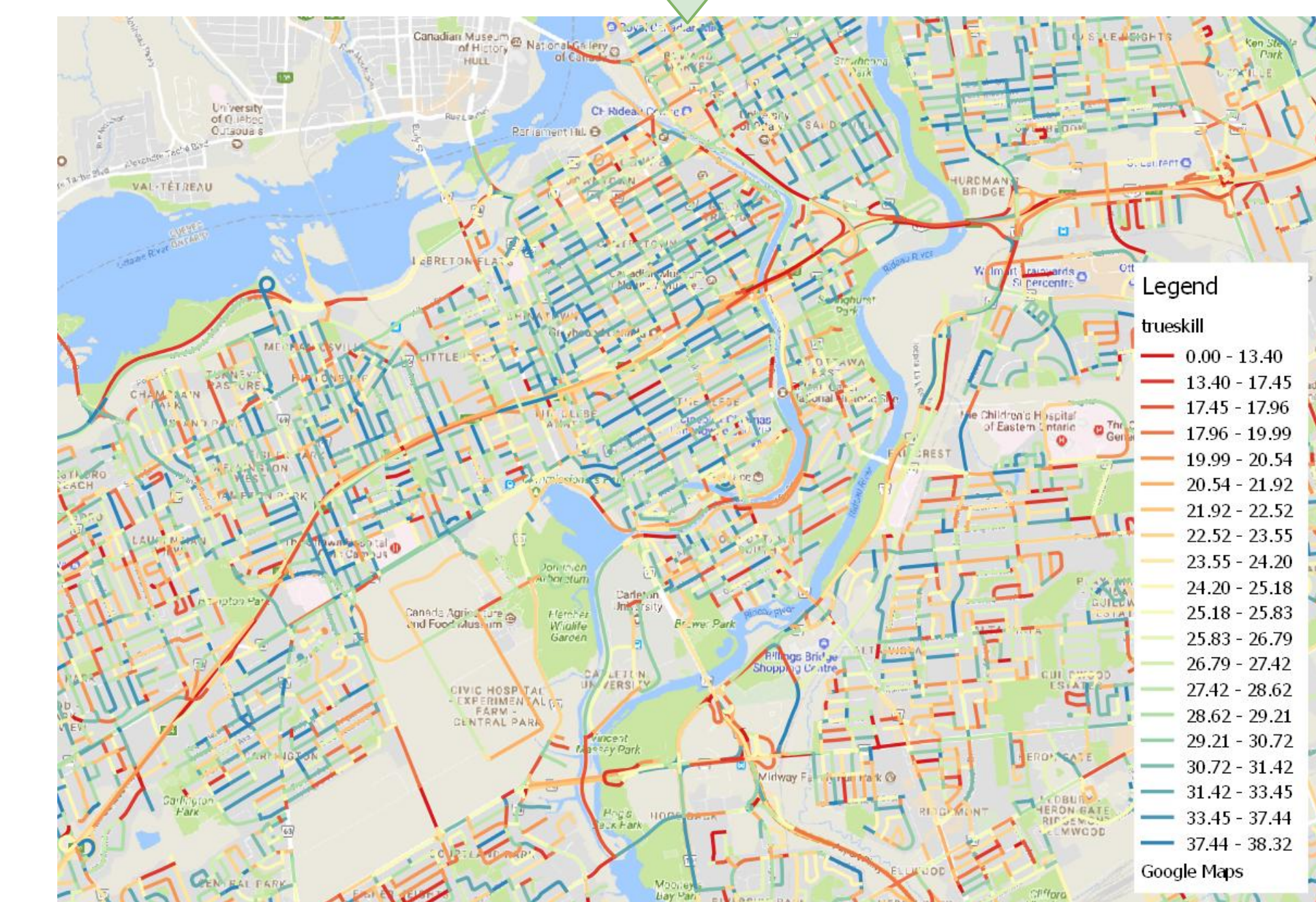
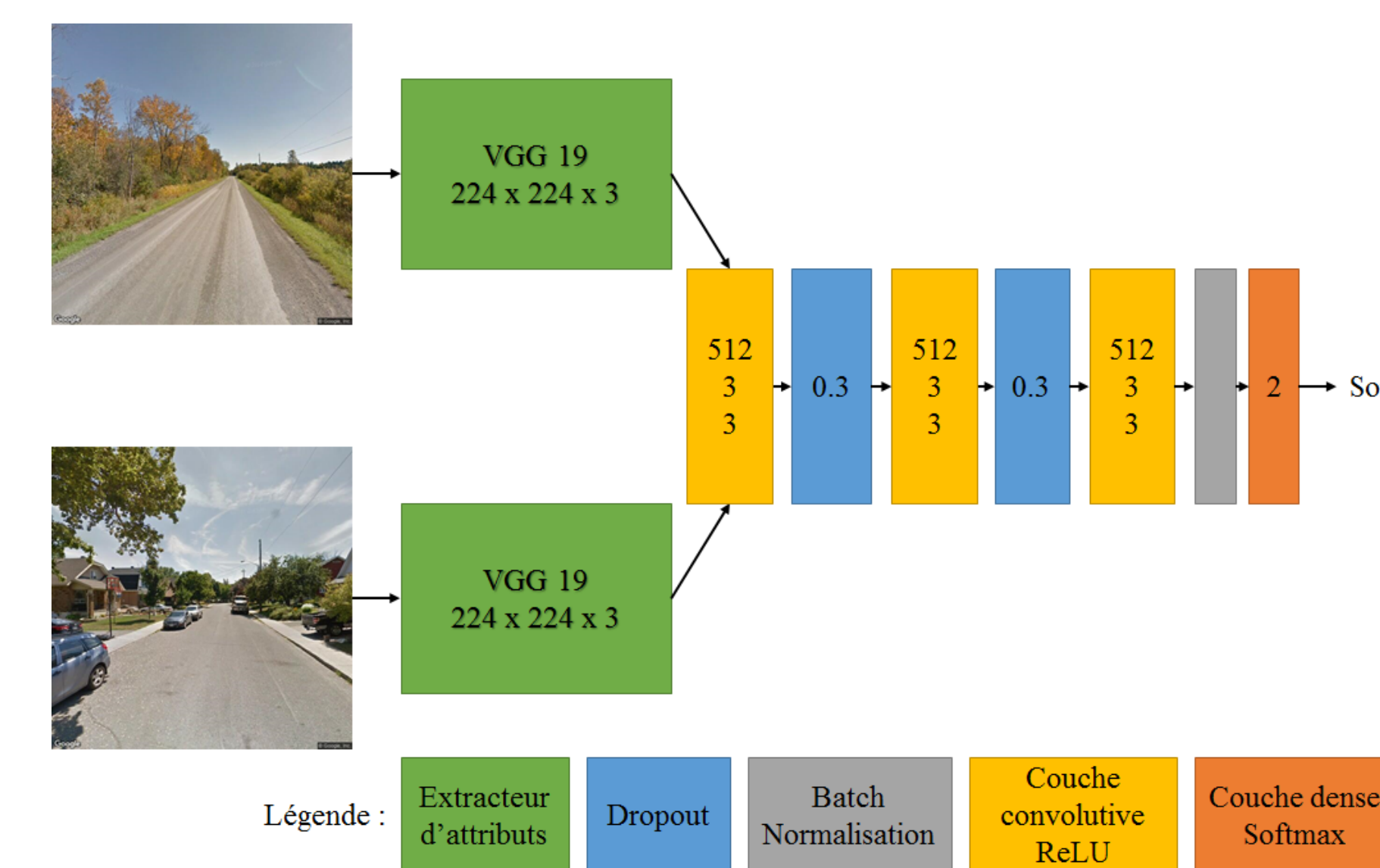


Figure 2.b : ComparaisonCroutiNET + Microsoft Trueskill

Figure 2 : Cartes de l'accessibilité urbaine du centre-ville d'Ottawa

Ces cartes (Figure 2) sont réalisées en calculant le score d'accessibilité urbaine de 25 000 images réparties dans la ville d'Ottawa (une image par tronçon de rue). Pour les deux architectures, on obtient des résultats similaires avec une forte corrélation. Sur les figures 2.a et 2.b, on peut remarquer que le centre-ville est l'endroit d'Ottawa avec la plus haute accessibilité urbaine. De plus, on distingue sur les deux représentations en rouge/orange (signe d'une mauvaise accessibilité urbaine) la 4x4 voies qui traverse le centre-ville de l'est au sud-ouest.

## 3) Génération de fausses images StreetView :

A l'aide d'une architecture dite de réseaux de neurones rivaux de type DCGAN (ALEC et al. 2016), j'ai généré de fausses images StreetView. Basé sur le principe du policier contrebandier, ce genre de réseaux simule une compétition entre deux personnes antagonistes (voir figure ci-dessous). Pour "apprendre à dessiner" de fausses images StreetView, le contrebandier (générateur) s'est entraîné sur 25 000 images.

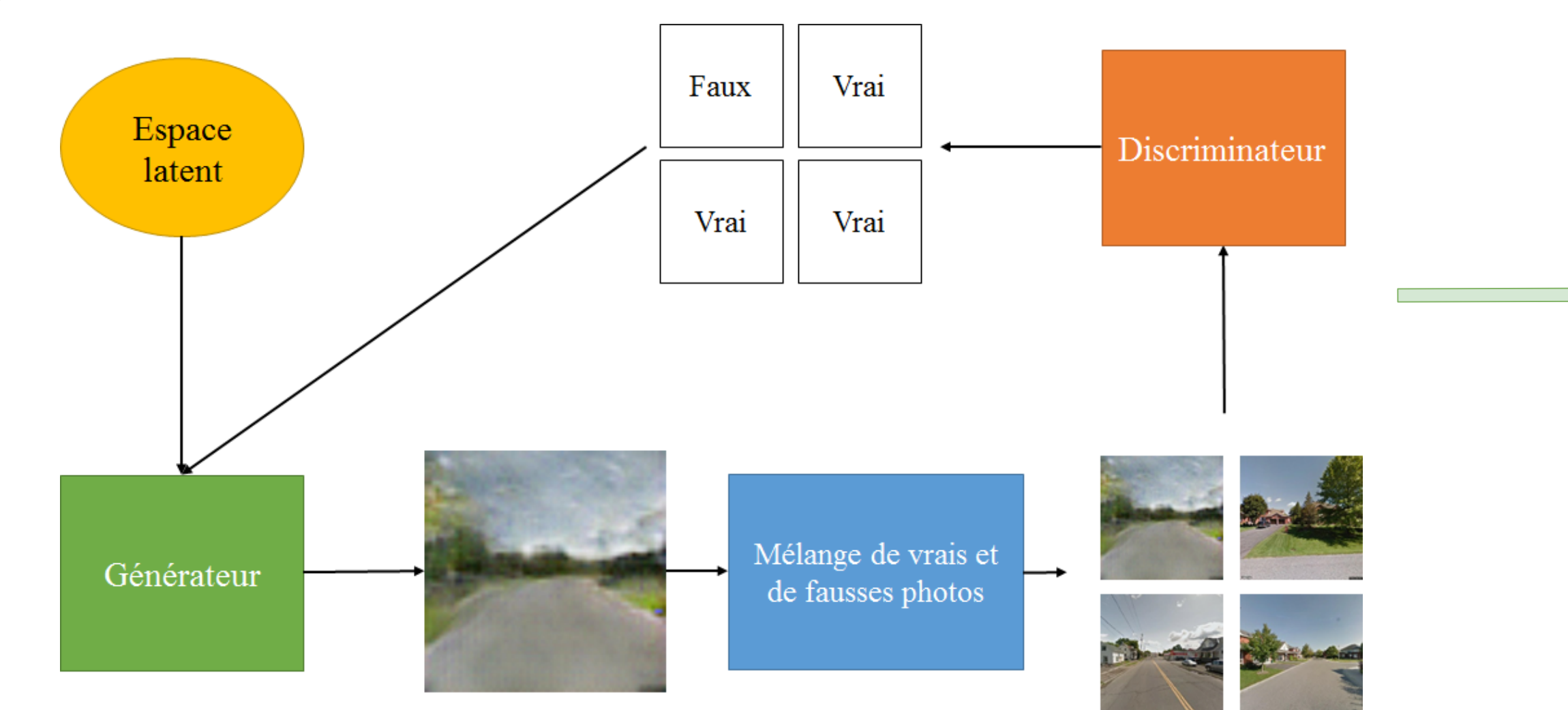
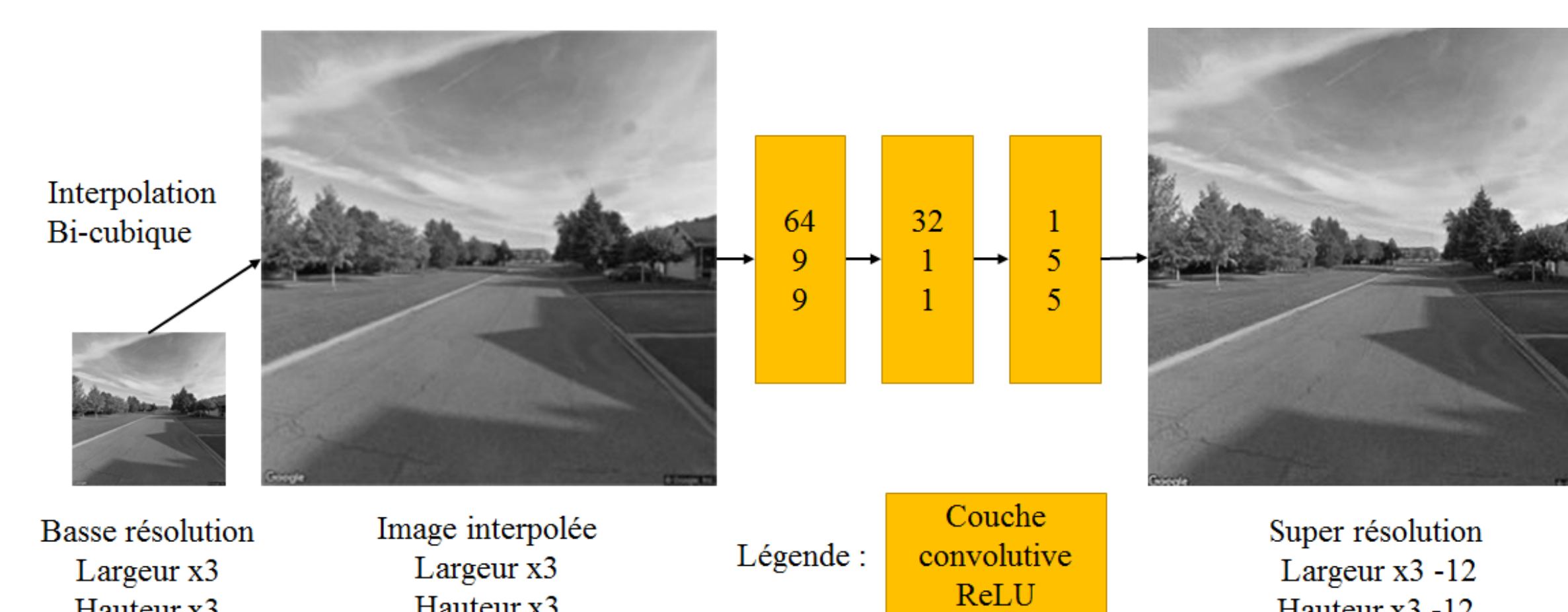


Figure 3 : Exemples d'images générées par DCGAN-CroutiNET

## 4) Augmentation artificielle de la résolution :

Dans un objectif de rendre plus réaliste les images générées par DCGAN, je me suis intéressé à l'augmentation artificielle de résolution d'images par deep-learning. Le principe de la super résolution est basé sur l'injection de détails de haute résolution dans une image basse résolution. J'ai repris l'architecture DONG et al. 2015 pour construire Super-Croutinet, un réseau qui fonctionne comme le montre la figure 4 :



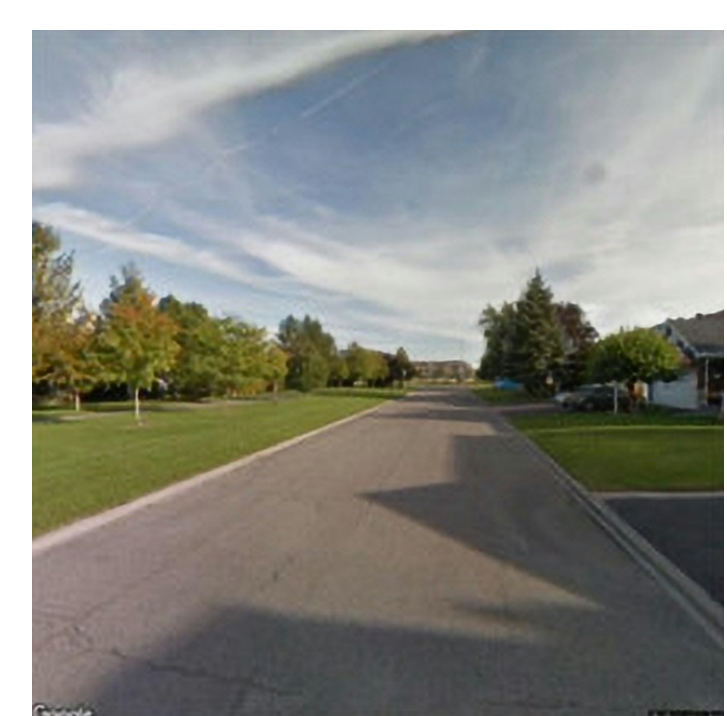
5.a : Basse résolution



5.b : Haute résolution



5.c : Interpolation



5.d : Super-résolution

Figure 4 : Architecture de SuperCroutiNET