Proposition détaillée

PSC INF02

Inférence dense de profondeur à partir d'une image de caméra monoculaire par apprentissage profond

Composition du groupe

Nom Prénom	Filière du concours
Le Hénaff Pablo	UNI
Zhang Jianfei	EV
Huang Zuli	EV
Leroux Antonin	MPI
Nivaggioli Adrien	PT
Dai Yayun	EV

Introduction

Le problème consistant à déterminer la profondeur réelle d'une scène photographiée est actuellement et ceci depuis quelques année activement étudié. Cela s'explique par la diversité des applications envisageable pour une telle technologie : les véhicules autonomes (sans conducteur), la détection et la reconnaissance d'objets, et tout ce qui nécessite une certaine autonomie de la part d'un robot/logiciel. En effet, dans une époque où la recherche s'acharne à rendre les véhicules, les robots, de plus en plus autonomes, il est capital que ceux-ci soient capables de reconnaître l'environnement qui les entourent, et plus particulièrement de se déplacer dans celui-ci. L'interêt de déduire la carte de profondeur à partir d'une ou plusieurs image(s) est que dans le cas d'un système embarqué il suffit d'utiliser un simple appareil photo plutôt qu'un ensemble de capteurs plus complexes permettant de déterminer exactement la profondeur.

Ce n'est donc pas étonnant que depuis de nombreuses années, des chercheurs tentent de mettre au point des algorithmes capables de déterminer la profondeur d'une scène photographique donnée.

Seulement, deux problèmes majeurs apparaissent : les données nécessaires et le temps de calcul. Bien souvent, un algorithme nécessitant moins de donnée va nécessiter moins de temps pour déterminer la profondeur, mais il risque d'être moins précis. Le système sur lequel l'algorithme est embarqué peut également ne pas être capable de collecter les données nécessaires. L'objectif est donc, aujourd'hui, de minimiser le nombre de données nécessaires, de maximiser les performances de notre algorithme, tout en optimisant celui-ci afin qu'il soit le moins gourmand possible.

Jusqu'à présent, les techniques les plus fructueuses nécessitaient plusieurs images d'un même lieu, afin d'obtenir un champ de profondeur correct. On rencontrait donc des méthodes dites de "stéréoscopie", qui travaillaient sur deux (ou plus) images de la scène

prises sous des angles différents, utilisant des techniques de triangulation, voir des méthodes exploitant le focus/défocus d'une caméra. On envisage aisément les difficultés à embarquer ces méthodes.

Il existe néanmoins des méthodes dites "monoscopiques", qui ne demandent qu'une seule image pour fonctionner. Grâce à un simple appareil photo, on peut ainsi collecter les données nécessaires. Différentes techniques existent déjà, que nous détaillerons plus dans la partie **I. État de l'art**, mais remarquons néanmoins que c'est un secteur en constante évolution (certains articles datant de septembre 2016), et que les performances augmentent à une vitesse phénoménale.

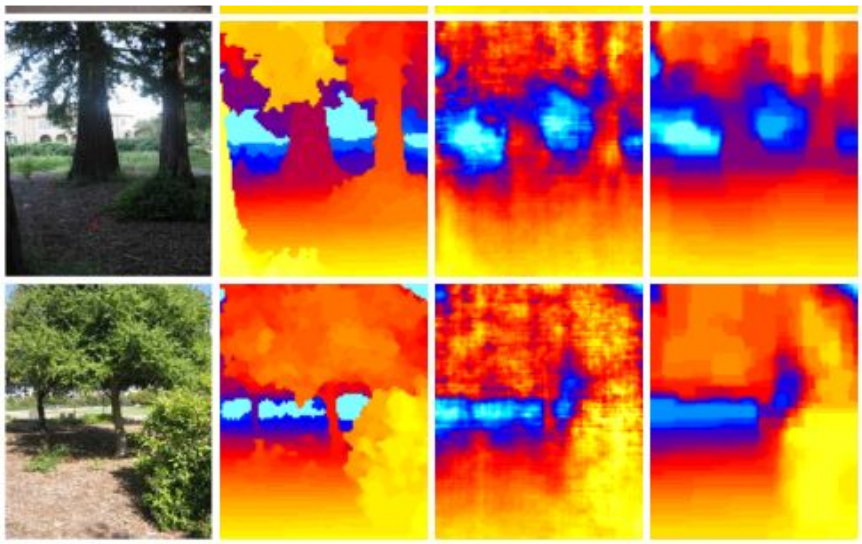
Pourquoi cette soudaine amélioration ? Bien que les techniques utilisées, basées sur des réseaux de neurones, datent des années 80, ce n'est que depuis très récemment que nous disposons de capacités de calcul nécessaire au bon fonctionnement de ces méthodes. Ce PSC propose donc, dans un premier temps, d'implémenter un réseau de neurone déjà existant afin d'obtenir des résultat proches de ceux de l'état de l'art, puis de tenter d'améliorer les performances, en établissant nous même de nouvelles approches. C'est un domaine "nouveau" (dans le sens où on ne peut expérimenter que depuis très récemment), et il reste encore énormément à découvrir. Nous sommes très enthousiastes à l'idée de faire partie des quelques équipes qui travaillent actuellement sur ce problème, et nous pensons pouvoir vraiment apporter quelque chose.

I. Etat de l'art

Faisons un petit historique des différentes méthodes utilisées pour résoudre les problèmes d'inférence de profondeur à partir d'une seule image.

Ce sont des problématiques que l'on commence à bien savoir résoudre depuis une dizaine d'années. Les premiers efforts en la matières ont majoritairement utilisé des approches probabilistes. L'idée était de s'appuyer sur des modèles probabilistes tels que les Markov Random Field (MRF) ou les Conditional Random Field (CRF). Le principe est de calculer la probabilité conditionnelle d'avoir une certaine carte de profondeur y, sachant qu'on a une certaine image x en entrée. Une fois cette probabilité calculée, il suffit de trouver la carte de profondeur la plus probable par rapport à l'entrée. Certaines fonctions proposées par la théorie des champs aléatoires fonctionnent plutôt bien pour répondre à ce genre de problèmes. Elles prennent en compte un certain nombre de paramètres telles que la texture et la couleur qui sont regardées sur des groupes de pixels, via des filtres. Une fois ceci fait une mise en relation des similitudes sur les différents groupes de pixels permet d'avoir une estimation de la profondeur de ces groupes de pixels (cf articles [4] et [5]).

C'est donc une première approche possible qui ne demande pas une trop grande puissance de calcul. Cependant bien que les résultats ne sont pas mauvais ils ne sont pas très précis. Comme on peut le voir avec l'image suivante qui montre le résultat d'un de ces algorithmes (les deux images les plus à droites sont à mettre en vis a des deux images les plus à gauche qui représentent la vérité terrain).

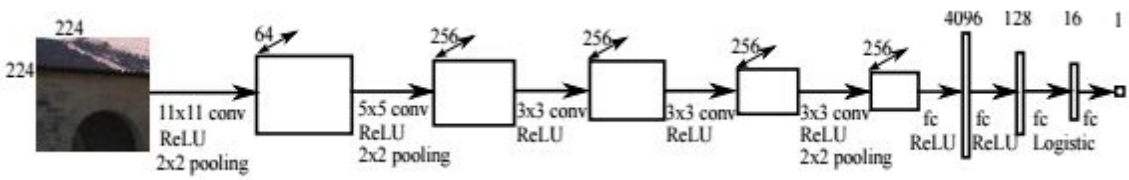


C’est pour cela que les chercheurs se sont tournés récemment vers des approches passant par du machine learning via des réseaux de neurones. Cette approche a concentrée la plupart des efforts ces 5 dernières années. Nous verrons que les résultats sont bien meilleurs, et les avancées technologiques ont rendu possible les calculs plus lourds que requièrent cette méthode.

Ces techniques s’appuient sur un outil très utilisé pour le traitement d’images et qui est adapté aux réseaux de neurones : les CNN (convolutional neural network). En effet, un des problèmes avec le traitement d’images est que l’on doit traiter une grande quantité de donnée (il faut prendre en compte tous les pixels). C’est pour cela que les CNN sont si utiles.

Revenons au principe de base des réseaux de neurones : on à une entrée sous forme d’un vecteur noté x et on veut avoir une sortie y qui est calculée à partir de x. Pour obtenir cette sortie, on multiplie ce vecteur par des matrices (chaque matrice représentant une couche). Normalement ces matrices ont des valeurs quelconques qui sont calculées pour optimiser le résultat par rapport à l’information qu’on veut obtenir en sortie. Dans le cas d’images, la taille du vecteur d’entrée est déjà énorme et faire tous les calculs matriciels est compliqué, alors que l’information pour chaque pixel est souvent contenue localement. C’est pour ça que pour construire un CNN on considère que cette matrice est non nulle seulement localement sur un petite zone dont on a défini la taille au préalable. Cela suffit pour avoir une information précise tout en limitant les calculs.

Un des premiers réseaux de neurones qui est apparu est le réseau AlexNet. Il obtient des résultats très bon avec une architecture assez simple.



On prend donc une zone de pixel de taille 224*224 et le but est d'estimer la profondeur du pixel au centre. On va donc appliquer une série de transformation de façon à réduire progressivement la taille des données jusqu'à atteindre l'information qu'on cherche : la profondeur. Chaque flèche indique le passage vers une nouvelle couche du réseau. Cela correspond simplement à la multiplication des données par une matrice et éventuellement une fonction de rectification.

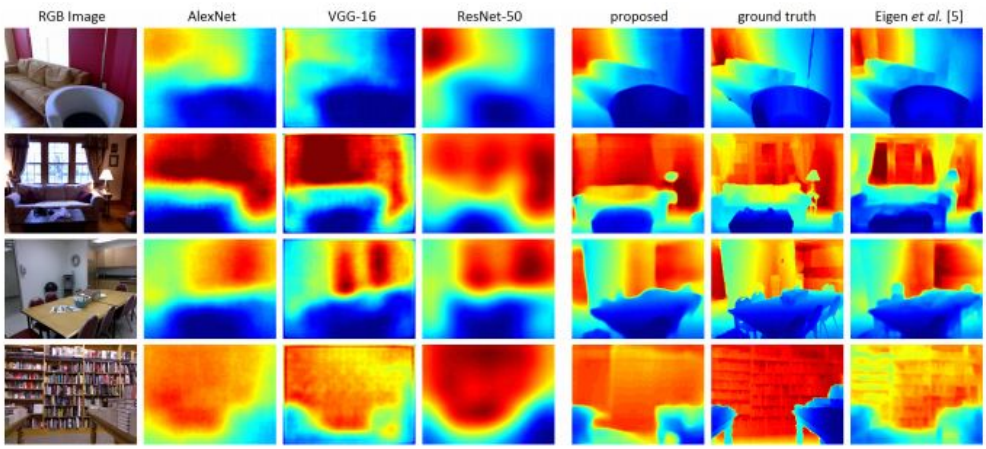
Quand on parle de fcl (fully connected layer) les termes de la matrice de transition sont tous non nuls. Ces couches-là n'apparaissent qu'à la fin car il faut que la taille des données ait été réduite pour éviter un calcul trop coûteux.

Les couches de x*x conv opèrent des convolutions : donc la matrice de transition est non nulle seulement sur une zone de taille x*x autour de chaque pixel.

Le pooling est une opération qui a pour but de réduire la taille des données traitées. Du 2x2 pooling revient à prendre des groupes de 4 pixels et les fusionner en un seul en prenant la plus grand valeur.

Enfin les termes ReLU et logistic sont les fonctions de rectification mentionnées précédemment. ReLU (rectified linear unit) supprime les termes négatifs qui n'ont pas de sens et les remplace par zéro. La transformation logistic a pour but de renvoyer toutes les les valeurs entre 0 et 1.

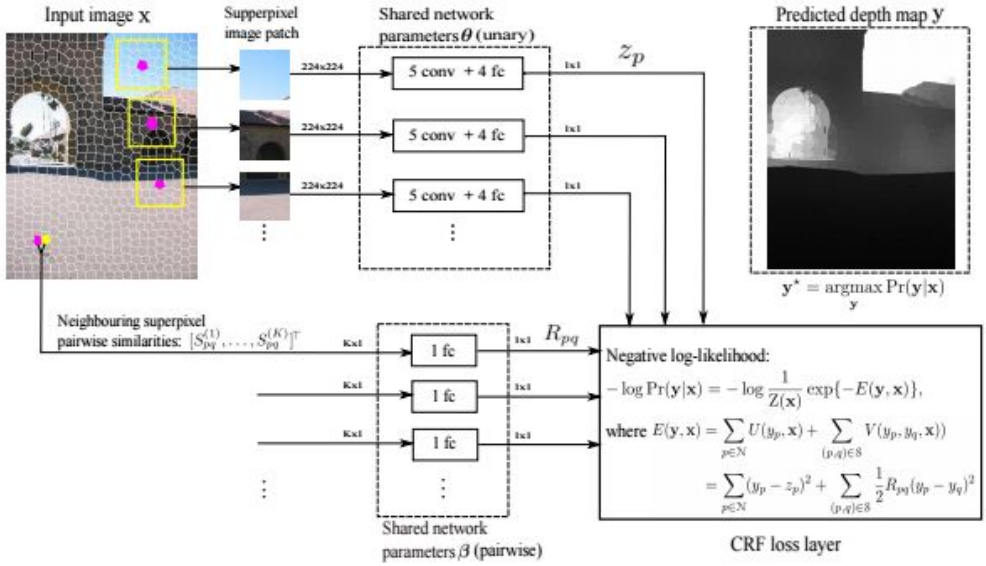
Voici quelques exemples de résultats fournis par des réseaux de neurones tirés d'un article de septembre 2016.



Les trois premières colonnes présentes des algorithmes assez simples de réseaux de neurones : AlexNet ainsi que VGG et ResNet (deux autres architectures de réseaux assez simples). L'avant dernière colonne présente la vérité terrain. Les deux colonnes restantes (proposed [1] ainsi que Eigen et al. [3]) présentent des résultats bien meilleurs et qui sont en fait très proches des réalités terrains que l'on obtient grâce aux scanners actuels.

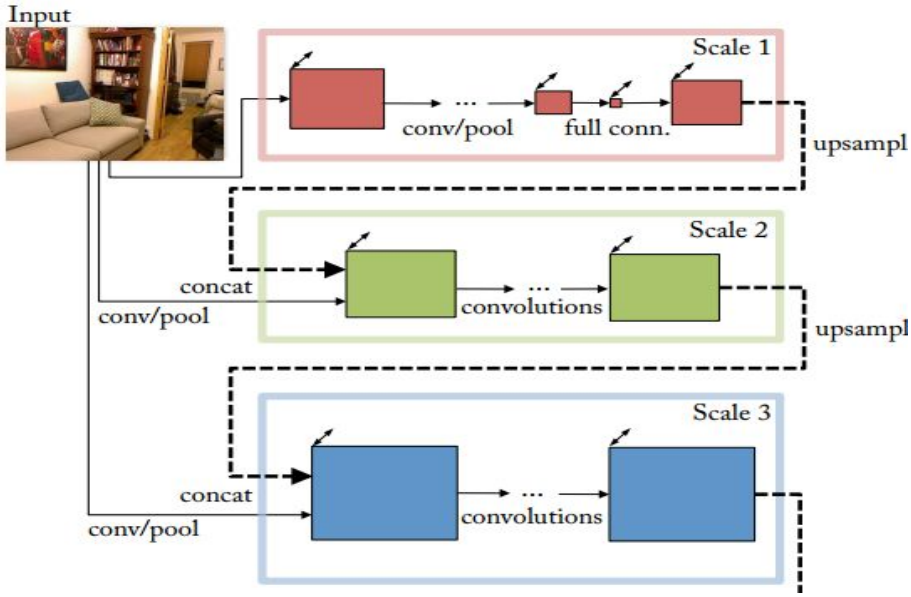
Les derniers articles sortis sur le sujet présente des réseaux beaucoup plus efficaces qu'un AlexNet. Les équipes qui ont écrit ces articles présentent ainsi des variantes de ces réseaux classiques.

L’équipe de Liu [2] propose ainsi le réseau suivant :



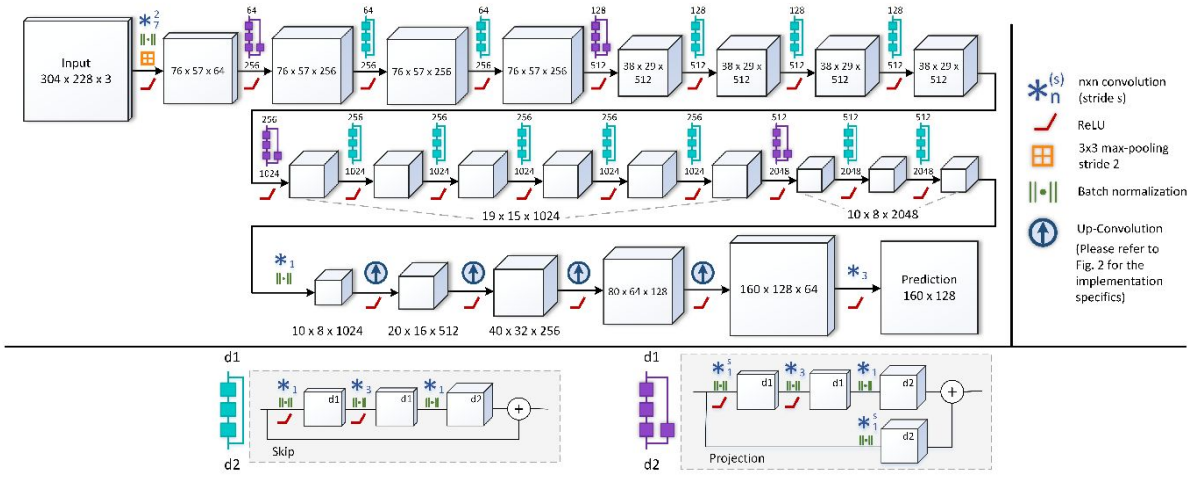
La branche du haut revient simplement à passer tous les blocs de pixel à travers l’AlexNet que nous avons montré précédemment. A ceci on ajoute un second réseaux de neurones assez simple afin de prendre en compte les similitudes entre les différents pixels (deux pixels avec des couleurs et des textures similaires peuvent avoir aussi des profondeurs similaires). Cet article propose un lien entre réseau de neurone et modèle probabiliste puisque la dernière partie consiste à maximiser une probabilité dont la formule vient tout droit de la théorie aléatoire des champs.

D’autres articles [2] propose une approche purement basée sur les réseaux de neurones mais en essayant de combiner ce problèmes avec d’autres tel que la reconnaissance de formes et le calcul de normal à des surfaces. Cela donne le réseau suivant :



Dans ce genre de réseaux souvent la première étape est un CNN classique tel qu'un AlexNet ou un VGG et ensuite on rajoute plusieurs réseaux (ici 2 autres) qui permettent d'affiner les résultats et d'augmenter la résolution de la sortie. L'idée ici est de créer un réseau polyvalent qui puisse traiter plusieur problèmes liés entre eux mais qui ne se calcule pas de la même façon, c'est typiquement vers ce genre de chose que se dirige les recherches actuelles sur l'intelligence artificielle: réussir à faire le lien entre différentes informations.

Une autre architecture de réseau est présenté dans l'article [1] :



Ici l'idée est de réadapter le réseau Res-Net 50. Les dernières couches fully connected sont remplacées par des couches de up-sampling créées par l'équipe qui a rédigé cet article.

Sur le tableau suivant qui est tiré de ce même article on retrouve les performances de tous les algorithmes les plus récents. Dont [2] et [3].

NYU Depth v2	rel	rms	rms(log)	$\log_{10}$	δ_1	δ_2	δ_3
Karsch *et al.* [10]	0.374	1.12	-	0.134	-	-	-
Ladicky *et al.* [15]	-	-	-	-	0.542	0.829	0.941
Liu *et al.* [20]	0.335	1.06	-	0.127	-	-	-
Li *et al.* [16]	0.232	0.821	-	0.094	0.621	0.886	0.968
Liu *et al.* [19]	0.230	0.824	-	0.095	0.614	0.883	0.971
Wang *et al.* [37]	0.220	0.745	0.262	0.094	0.605	0.890	0.970
Eigen *et al.* [6]	0.215	0.907	0.285	-	0.611	0.887	0.971
Roy and Todorovic [27]	0.187	0.744	-	0.078	-	-	-
Eigen and Fergus [5]	0.158	0.641	0.214	-	0.769	0.950	**0.988**
ours (ResNet-UpProj)	**0.127**	**0.573**	**0.195**	**0.055**	**0.811**	**0.953**	**0.988**

L'article [1] a clairement les meilleurs résultats dans tous les domaines. C'est donc l'algorithme proposé dans cet article que nous avons choisi d'implémenter avec nos tuteurs.

II. Organisation du travail et répartition des tâches

Une première équipe (2 personnes) sera chargée de collecter des données. Ces données sont disponibles sur internet : nous essaierons de travailler sur le même jeu de données que les quelques autres équipes internationales. Cela nous permettra de comparer nos résultats de manière objective lors de la phase d'amélioration. Par ailleurs, cette même équipe devra apporter les outils nécessaire à l'exploitation de ces données : notamment programmer des algorithmes de lecture des images et des 'vérités terrains" c'est-à-dire des données de profondeur collectées par le scanner accompagnant la photographie. Cette équipe sera aussi chargée d'imaginer et développer des outils de mesure des performances des algorithmes que nous allons exécuter. Il faudra s'intéresser aux mesures de performance utilisées dans les publications scientifiques afin d'avoir des références communes.

En parallèle, un autre groupe (auquel s'ajouteront les membres de la première équipe si leur tâche est plus rapide) devra appréhender l'utilisation de Torch. Quelques tutoriels sont disponibles en ligne. Puis il faudra récuperer un réseau de neurones déjà appris.

Un membre du groupe effectuera une veille des nouveaux articles scientifiques publiés dans le domaine.

Nous avons dans le groupe une grande diversité d'expériences puisque nous avons suivi des cursus très différents avant le concours de l'École polytechnique. C'est pourquoi certains d'entre nous sommes plus habiles pour certaines tâches (notamment programmation). Il faudra composer les différents groupes en fonction.

Une réunion hebdomadaire permettra à chacun de mesurer l'avancée du travail des autres membres du groupe.

III. Matériel requis

Pour tester et entraîner notre propre réseau de neurones nous aurions besoin de l'accès à des GPU. L'École dispose des moyens nécessaires, il faut pour y accéder contacter le laboratoire d'informatique.

IV. Objectifs intermédiaires - avec leur échéancier

Phase 1 : Apprentissage des outils et de la théorie du Deep Learning / *Septembre 2016 - Octobre 2016*

Nous suivons notamment les cours en ligne à propos du Deep Learning afin d'avoir les connaissances fondamentales.

Phase 2 : Recherches d'articles de la littérature dans le domaine de l'inférence de profondeur issue d'images monoculaires / *Octobre 2016 - Novembre 2016*

Nous lisons plusieurs articles dont les méthodes proposées sont différentes pour obtenir une vue complète des approches d'inférence de profondeur.

Phase 3 : Mise en place de l'algorithme que nous aurons choisi par rapport aux articles proposés pendant la phase 2 / *Novembre 2016 - Janvier 2016 (date du rendu du livret intermédiaire)*

Phase 4 : Entraînement du modèle en fonction des RGB-D data-sets tels que NYU Depth / *Janvier 2016 - Février 2016*

Nous utilisons plusieurs datasets pour entraîner les paramètres du modèle.

Phase 5 : Évaluation de la performance de l'algorithme, amélioration du modèle avec des propositions pertinentes / *Février 2016 - Mars 2016*

Nous évaluons le modèle par test data-set. Selon ce que nous observons, nous améliorons pour qu'il s'adapte mieux aux appareils moins puissants et ait une meilleure performance.

Phase 6 : Implémentation de l'algorithme amélioré, évaluation de performance / *Mars 2016 - Avril 2016*

Nous mettons en place le modèle amélioré et rédigeons le rapport final.

Conclusion

Comme nous l'avons vu, la vision 3D rassemble tout un ensemble de technologies améliorant les performances des produits optroniques mais permettant aussi d'apporter une meilleure expérience utilisateur aussi bien en termes de compréhension du contexte que pour des problématiques d'interaction homme-machine. Les technologies passives d'estimation de profondeur sont les plus discrètes pour opérer sur un théâtre d'opération. En tirant parti du mouvement du système optronique, les technologies actuelles permettent notamment des reconstructions 3D denses et dans de grands environnements. Cependant ces technologies échouent lorsque le système optronique est immobile ou que d'autres conditions ne sont pas remplies (scène peu structurée distances non compatibles, configurations dégénérées, ...).

Dans ce contexte, notre projet consiste à étudier les méthodes d'inférence de profondeur à partir d'une seule image de caméra monoculaire. Depuis une dizaine d'années les méthodes qui utilisent les modèles probabilistes tels que les Markov Random Field (MRF) ou les Conditional Random Field (CRF) ont été explorées. Cette première approche ne demande pas une trop grande puissance de calcul mais les résultats obtenus ne sont pas très précis.

Grâce au développement rapide de la puissance de calcul de l'ordinateur, le « deep learning » s'est imposé comme une rupture technologique dans le domaine du « machine learning » et il a été appliqué aux domaines variés avec des résultats assez remarquables. Plus particulièrement, avec la présence de grandes bases de données, l'apprentissage profond permet de franchir un gap de performance dans le domaine de reconnaissance d'image, y compris l'inférence de profondeur que l'on étudie dans ce projet. Les méthodes proposées basent sur les réseaux convolutionnels profonds, qui tiennent compte mieux des informations locales d'une image par rapport des réseaux neurones classiques.

Donc, dans un premier temps, nous allons implémenter un réseau de neurone présenté dans [1] afin d'obtenir des résultats proches, puis de tenter d'améliorer les performances, en établissant nous-même de nouvelles approches. L'existence des bibliothèques de l'apprentissage profond tel que Torch que l'on va utiliser pour ce projet facilite l'implémentation des réseaux de neurone consistant en plusieurs couches. L'accès à des GPU puissants et des algorithmes optimisés accélèrent le calcul.

Outre les connaissances scientifiques dans ce domaine majeur de notre projet (apprentissage profond), ce PSC nous permettra d'apprendre à travailler au sein d'une équipe dans un domaine moderne en plein développement.

References

[1] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. *arXiv preprint arXiv:1606.00373, 2016*.
https://arxiv.org/pdf/1606.00373v2.pdf

[2] Liu, F., Shen, C., & Lin, G. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 5162-5170, 2015.
https://arxiv.org/pdf/1411.6387v2.pdf

[3] Eigen, D., & Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision,* pages 2650-2658, 2015.
https://arxiv.org/pdf/1411.4734v4.pdf

[4] Saxena, A., Chung, S. H., & Ng, A. Y. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems,* pages 1161-1168, 2005.
https://papers.nips.cc/paper/2921-learning-depth-from-singlemonocular-images.pdf.

[5] Saxena, A., Sun, M., & Ng, A. Y. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, *31*(5), 824-840, 2009.
http://www.cs.cornell.edu/~asaxena/reconstruction3d/saxena_make3d_learning3dstru
cture.pdf