Université de Caen

Master imalang

Rapport : Stage de fin d'études

Vision par Ordinateur

Segmentation, Extraction et Reconnaissance Visuelle d'Éléments dans des Contextes Variants

---

Tuteur Professionnel :
Arnaud Saval

Tuteur Pédagogique :
Youssef Chahir

Stagiaire :
Valentin Laforge

mars – septembre 2016

Résumé

Au cours des dernières décennies, les outils et les services mis à disposition du grand public ont provoqués l'apparition de larges bases d'images. L'ampleur toujours grandissante de celles-ci a donnée naissance à la nécessité de classifier les images de façons automatiques, rapides et non supervisées. Parallèlement, des robots intelligents et des systèmes de reconnaissances de postures commencent à faire leurs apparitions dans nos foyers tandis que nos véhicules embarquent de plus en plus d'outils d'assistance à la conduite pour enfin devenir autonomes. Ainsi, la communauté scientifique et l'intérêt que suscite la vision par ordinateur n'ont cessés de croître. Des concours sont organisés et mettent à épreuves les approches de reconnaissances d'objets, de sémantique, de scène, d'émotions et d'actions dans les images. Aujourd'hui, les propositions mènent à des résultats suffisamment fiables pour des utilisations permanentes sur des cas complexes. Les systèmes répondent à de nouvelles problématiques ou succèdent aux systèmes de classification existants. Ce rapport en décrit les solutions qui s'y portent et l'implémentation de systèmes de vision par ordinateur pour l'aménagement technologique des entreprises et des outils des professionnels.

Remerciements

Pour commencer, je tiens à remercier vivement Arnaud Saval pour avoir rendu ce stage possible, pour m'avoir attribué toutes les ressources nécessaires afin d'en faire une réussite et surtout pour m'avoir orienté entre les différentes problématiques de façon formatrice.

Je remercie également Arthur Vaisse Lesteven, Clément Caron et Guillaume Debras, pour m'avoir encadrés et facilité mon intégration dans l'entreprise.

Je remercie particulièrement Esther Nicart ainsi que Youssef Chahir pour nos discussions fructueuses autour du machine learning et des tâches de segmentation.

Enfin, je tiens à remercier l'ensemble des employés de Cordon DS2I pour leur accueil et je salue Guillaume Leroy, le second stagiaire de l'équipe innovation, avec qui j'ai partagé mon bureau pendant les deux derniers mois de mon stage, pour son aide et sa bonne humeur.

Table des matières

Rapport de stage

1 Introduction

1.1 Contexte

Le présent rapport fait état du **stage** de fin d'études réalisé par l'étudiant Laforge Valentin dans le cadre du deuxième semestre de dernière année en Master Informatique IMALANG (Traitement automatique de l'image et du langage du texte). Ce stage, d'une durée de 6 mois, supervisé par le tuteur professionnel Saval Arnaud et le tuteur pédagogique Chahir Youssef s'inscrit dans la spécialisation du traitement automatique de l'image et concerne l'implémentation de systèmes de vision par ordinateur. Certains aspects du stage sont soumis à confidentialité et ne seront, par conséquent, pas mentionnés ou vulgarisés dans ce document.

L'**organisation du rapport** a été conçue de manière à se montrer représentatif de l'évolution des travaux du stagiaire sur les problématiques étudiées. D'un choix complètement arbitraire, on accorde donc une attention particulière aux approches méthodiques des travaux réalisés plus qu'au fonctionnement purement technique des implémentations. De même, une part importante des recherches réalisées pendant le stage et qu'il ne faudra pas négliger à la lecture de ce rapport se présente en tant que document indépendant sous forme d'État de l'Art, en Annexe 1. Enfin, en raison de la différence entre les dates de fin de stages et celle de la remise du rapport pour évaluation auprès de l'Université, les quelques dernières semaines de stage ne pourrons être traitées.

1.2 Organisme d'accueil

1.2.1 Cordon DS2I

Cordon DS2I (**Défense**, **Sécurité**, **Imagerie** & **Innovation**) développe des systèmes d'informations experts, dans le domaine du renseignement, de la surveillance et de la sécurité. Cet ancien site Airbus situé à Val-de-Reuil, repris le 1er Octobre 2015, travaille notamment dans la télécommunication militaire, le traitement de l'information géolocalisée et de données multimédia, l'imagerie, l'intégration de véhicules et shelters de tous types, la sécurité des personnes et des institutions à travers le monde.

1.2.2 Historique

Cordon DS2I, situé à Val-de-Reuil et récemment repris par Serge Cordon, étais antérieurement un site de Cassidian (formé en 2010, précédemment EADS), appartenant à Airbus Defense & Space. Si les secteurs d'activités du site de Val-de-Reuil restent inchangés après le rachat, la répartition des employés et des projets ont été altérés, l'historique de Cordon Electronics est le suivant :

- **2015** : Reprise d'un centre Alcatel-Lucent à Orléans (Cordon Networks), Ouverture d'un centre en Allemagne, Reprise d'un centre AIRBUS Defence & Space près de Rouen (Cordon DS2i)
- **2014** : Création de CORDON CMS (ancien centre SAV SONY en Alsace)
- **2013** : Création de CORDON DO BRASIL à Curitiba
- **2012** : Création de CORDON REUNION et de CORDON CONTACT
- **2011** : Rachat de TR2S, filiale SAV de Sagemcom
- **2010** : Démarrage des prestations sur les modems ADSL
- **2009** : Acquisition de S2MI (groupe Carrefour), réparation de PC
- **2008** : Rachat du SAV Thomson Europe
- **2005** : Rachat de Cirma-Cegelec à Bordeaux
- **2004** : Acquisition de sociétés SAV en Roumanie et aux Caraïbes
- **2003** : Reprise des activités de SAV Mitsubishi Mobile
- **2000** : Création d'une filiale en Hongrie. Rachat des activités Matra Telecom à Rennes
- **1995** : Démarrage de la réparation de téléphones portables
- **1989** : Création par Mr Serge CORDON

1.2.3 Accès au site

Les employés du site de Val-de-Reuil travaillent sur des sujets classés *Secret Défense* et *Confidentiel*, les accès aux différentes zones et bâtiments sont, par conséquent, restreints et surveillés. Les employés ne possèdent pas tous les mêmes autorisations et des tourniquets à badge limitent les accès. Le badge de stagiaire justifie sa présence sur le site et s'il n'ouvre que les portes du bâtiment dans lequel il travail, ce dernier doit néanmoins être accompagné en permanence par au moins un de ses encadrants. Enfin, un outil de l'entreprise lui permet de pointer pour attester de sa présence.

1.2.4 Équipe Développement et Intégration

Le stage s'est déroulé au sein du pôle développement et intégration de Cordon DS2I. L'équipe, composée d'informaticiens d'origines diverses (Thésards, Ingénieurs ...) travaille sur l'étude et la réalisation de projets informatiques. Ceux-ci répondent à des problématiques nouvelles et sensibles dans l'optique de proposer des solutions innovantes à ses clients ou

d'améliorer les systèmes existants. Curieusement, il ne s'agit pas pour autant du pôle R&D de Cordon Electronics.

1.2.5 Cadre de travail

Le stage étant orienté Vision par ordinateur, le travail fut réalisé sur des postes informatiques présents dans un espace ouvert de développement. Les outils dédiés et mis à disposition du stagiaire étaient alors les suivants :

- Différents **systèmes d'acquisition** tels qu'une Webcam, une caméra de surveillance, une Kinect ;
- Différents **postes informatiques** :
  - *Ubuntu 14.04.1 double écran avec accès administrateur et webcam ;*
  - *Ubuntu 14.04.1 écran simple avec accès administrateur et Kinect ;*
  - *Windows 7 sur le réseau interne de l'entreprise (boite mail, pointage) ;*
- L'accès à un **serveur** *Proliant Gen8 HP* disposant de plus de 80Go de mémoire vive et de 32 coeurs cadencés à 2.4 GhZ.

Enfin, l'acquisition de ressources matérielles et intellectuelles étant offerte, aucune contrainte ne vint à l'encontre du bon déroulement du stage.

Les horaires du stagiaire sont flexibles, ils sont laissés à sa discrétion. Celui-ci gère son temps à sa convenance pour s'acquitter de son volume horaire journalier s'élevant à sept heures. L'entreprise propose un système de restauration de qualité à prix avantageux ainsi que d'un parking dédié.

1.3 Objectifs du stage

Au cours du stage et en accord avec le développement des projets concernés, les objectifs ont évolué. Ces objectifs tels qu'initialement proposés et décrits par l'entreprise sont les suivants :

L'objectif du stage est de développer un module de **reconnaissance visuelle** basé sur différentes features. Une étude sur ces dernières devra être réalisée, de manière à les étudier et comparer afin de proposer un algorithme adaptatif permettant d'utiliser à la volée les features adéquates aux images visibles.

Une étude concernant la manière de gérer l'apprentissage, ainsi que la base de données utilisée, devra être effectuée. Une première idée concernant la mise en place d'une hiérarchie des objets en fonction de type pré-définis par des experts est envisagée et sera amenée à évoluer au fil du stage.

Le module ainsi créé sera intégré au sein de la plateforme Cordon DS2i.

Les principales tâches du stage seront :

- L'assimilation de l'existant, du contexte, des besoins et contraintes Cordon DS2i ;
- L'état de l'art sur la thématique (reconnaissance de formes, descripteurs, indexation visuelle, etc.) ;
- L'étude/recherche algorithmique ;
- L'implémentation d'un démonstrateur (Java, C++, ...) et d'un wrapper R.O.S avec validation et évaluation des performances sur un cas réel ;
- La rédaction d'un rapport d'étude.

Afin de mener à bien ces tâches, vous travaillerez en collaboration avec votre tuteur et les autres membres de l'équipe Innovation, ainsi qu'avec des ingénieurs d'autres filiales du groupe.

Nous verrons plus tard, dans le bilan de ce rapport et en relation avec les évolutions des projets concernés, les différences entre les tâches initialement conçues en amont du stage et les implémentations réalisées.

2 Vision par Ordinateur : Reconnaissance visuelle d'objets

Les sujets travaillés au cours du stage portent sur du traitement automatique d'images. L'automatisation du traitement est rendu possible grâce à un ordinateur. On parle alors de Vision par ordinateur. Dans cette partie, nous décrirons les méthodes et les approches ; le détail des implémentation, notamment l'aspect technique est décrit plus tard, dans la partie 4.1.

2.1 Projet RECOB

Le premier projet auquel le stagiaire a pris part est le projet RECOB. Celui-ci a pour but d'améliorer un système existant de classification d'objets extraits d'une boîte. Ce projet concerne directement la reconnaissance d'objets électroniques (télécommandes, chargeurs, câbles Ethernet/Téléphone/d'interfaçage multimédia, modem, adaptateurs de prise téléphonique), ordonnés dans l'espace ou non par leurs utilisateurs lors d'un renvois des dispositifs au fournisseur.

2.1.1 Système existant et voies d'amélioration

Le système existant utilise l'homme en tant que vecteur de reconnaissance. Celui-ci ouvre une boîte, constate ou non la présence d'objets attendus qu'il classifie avant de passer à la suivante. L'estimation de l'erreur moyenne de l'humain pour cette tâche s'élève à 10%.

L'objectif de RECOB est de proposer une amélioration à ce système. Pour cela, il devra représenter un avantage économique qui sera amené par une réduction du coût moyen de traitement d'une boîte en temps et ressources. Nous estimerons ce premier coût actuel et total à 45 secondes. Les gains en ressources seront issus de l'endiguement des blessures provoquées par la manipulation d'objets coupants et des fatigues musculaires de manipulation.

2.1.2 Proposition RECOB

La solution proposée par le service Développement et Intégration est l'implémentation d'un système de vision par ordinateur. Une unité de calculs établira visuellement si les objets attendus sont présents ou non. Il pourra alors instantanément classifier ceux-ci pendant que l'employé s'occupera, en parallèle, de ranger les cartons et de préparer le suivant. Il pourra enfin valider les résultats de la reconnaissance de l'ordinateur avant de passer au carton suivant. Les avantages sont alors multiples, la reconnaissance a lieu plus rapidement, le processus est parallélisé (création de temps masqués), l'employé est moins fatigué (donc plus vif et attentif en fin de journée) et on espère que la validation des résultats de l'ordinateur par l'homme nous fera descendre en dessous des 10% actuel d'erreurs.

2.1.3 Recherche et Développement, État de l'art

Pour établir quelles seront les méthodes du système de vision implémentés, une phase de recherche a été réalisée. La nécessité de la rédaction de l'état de l'art pour la reconnaissance visuelle d'objets dans un flux vidéo s'est fait ressentir et a motivée le recrutement d'un stagiaire dont les études et la spécialisation correspondent au domaine. Ce document, qui vous est disponible en **Annexe [1]** constitue **une part essentielle de ce rapport** puisqu'il s'inscrit dans la continuité de l'enseignement Universitaire en traitement automatique de l'image reçu par l'étudiant et concrétise sa formation (je préconise donc fortement sa lecture). Il s'agit d'un état de l'art intéressé et rédigé avec le soucis des biens du projet RECOB à l'esprit. Il n'est donc pas complètement exhaustif et les méthodes jugées non applicables ne sont pas toutes indiquées ou bien se retrouvent moins approfondies. Il est néanmoins suffisamment complet et pertinent pour être utilisé dans un cadre différent. Enfin, il est voulu accessible pour quiconque souhaiterais le lire et tente de vulgariser les propositions dans le domaine de la reconnaissance visuelle en prenant soin de citer les articles concernés pour les lecteurs souhaitant prendre connaissance et approfondir le sujet.

2.1.4 Communauté, conventions et concours

La reconnaissance d'objets dans les images et les flux d'images est un domaine qui suscite l'intérêt de nombreux chercheurs et entreprises. En effet, bien que l'homme soit capable reconnaître les objets avec une précision d'environ 90%, le laisser classifier les images est une tâche trop coûteuse en temps et ressources. La problématique a reçu de nombreuses propositions et améliorations de méthodes de reconnaissance au cours des dernières décennies, notamment grâce aux conventions comme le *Computer Vision and Pattern Recognition (CVPR)* et les concours tels que le *(Pascal) Visual Object Classes (VOC)*[10] (2005-2012), le *Low-Power Image Recognition Challenge (LPIRC)* ou le *Large Scale Visual Recognition Challenge (LSVRC)* d'IMAGENET qui en rassemble les plus grands acteurs (GOOGLE et MICROSOFT entre autres).

Nous avons eu la chance d'assister à la journée sur le "Deep Learning", organisée par NORMASTIC (la fédération normande de recherche en sciences et technologies de l'information et de la communication) à l'université de Caen le 28 Juin 2016. Cet événement présentais particulièrement un tutoriel sur les réseaux de neurones ainsi que les travaux de chercheurs sur des problèmes de reconnaissances les exploitants.

2.1.5 Objets ciblés : étude de cas

La reconnaissance visuelle porte sur des objets électroniques. Ceux-ci sont les suivants : un ensemble de câbles (RJ45, RJ11, HDMI, alimentation) et de boîtiers (télécommande, bloc d'alimentation, modem, adaptateur de prise téléphonique).

2.1.5.1 Les boîtiers

Ce sont les éléments les plus facilement identifiables. Pour chaque type de boîtiers, la variabilité intra-classe est relativement faible tandis que la variabilité inter-classe est élevée. Les caractéristiques intéressantes de ces objets seront alors

— leurs formes,
— la présence d'autocollants et logo,
— leurs tailles ou surface (qui est bien plus importante que celles des câbles).

2.1.5.2 Les câbles

Ce sont les éléments les plus complexes. Leurs structures filaires implique une variabilité intra-classe conséquente pour plusieurs raisons :

— La forme globale de l'objet dépend de la façon dont le câble est noué, enroulé, jeté dans le carton,
— Tous les câbles RJ45 que nous aurons à traiter sont les mêmes (couleurs et forme des embouts, taille) mais les câbles RJ11 partagent très peu de points communs (seuls la présence d'une languette et la structure filaire ne varient pas).

2.1.6 Objets ciblés : orientation algorithmique

Parce que nous venons de voir que les objets sont très différents les uns des autres, l'approche à avoir pour la classification de chacun de ceux-ci doit être adaptée. Ainsi, chaque cas est sujet à être identifié par des méthodes différentes.

2.1.6.1 Câbles RJ45

Appartenant au groupe d'objets les plus complexes à identifier, le câble RJ45 possède des bouts d'une forme et d'une couleur invariantes dans le cadre du projet RECOB. Une première bonne approche est de réaliser de la reconnaissance couleur de l'objet pour en segmenter les bouts. On pourra alors passer par une deuxième reconnaissance des régions extraites au niveau de leurs formes ou de leurs textures afin de finalement valider la présence du câble.

Détection colorimétrique :

En travaillant sur le Hue Saturation Value (espace colorimétrique) (HSV) d'une image acquise, on peut chercher à isoler le jaune très caractéristique des embouts par une binarisation basée couleur. Les systèmes d'acquisition n'étant pas parfaits, les couleurs seront légèrement parasitées (présence de vert) et parce que la scène possède un éclairage orienté, les bords de l'objet peuvent éventuellement réfléchir la lumière ou être ombrés. Pour répondre à cela, la segmentation admet une tolérance d'écart à la valeur de référence choisie et bénéficie d'une ouverture morphologique. Avec ce système, nous pouvons extraire nos embouts en fonction de la visibilité de ceux-ci. En effet, il n'est pas robuste à l'occultation ; toutefois, il est très robuste à la translation, la rotation, au changement d'échelle, à la qualité d'acquisition et

même au flou. Nous ajoutons enfin de la flexibilité au système en prenant pour acquis que la présence d'un embout visible implique la présence d'un second, même occulté.

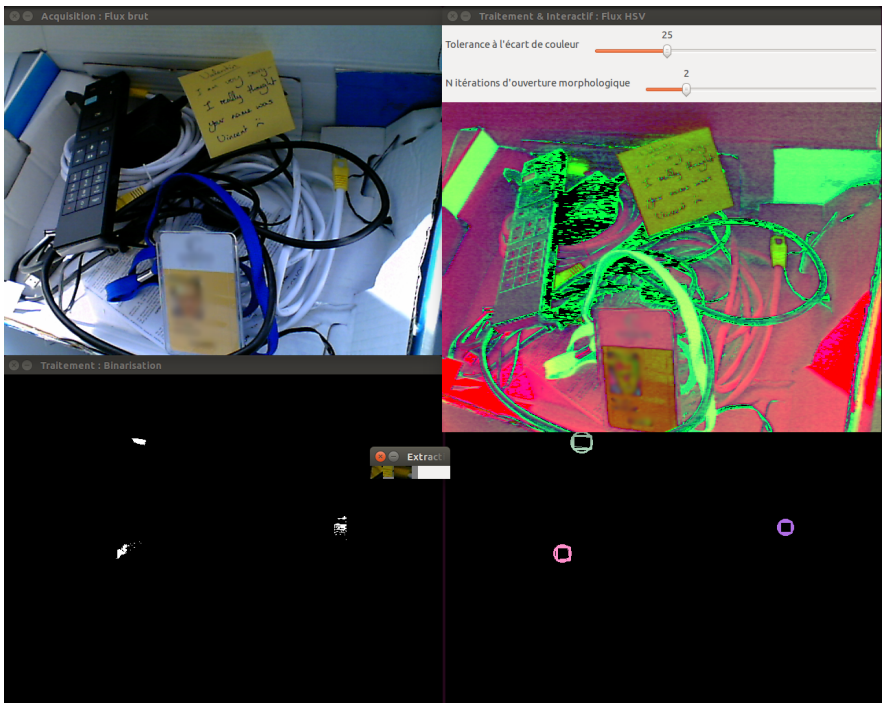


FIGURE 1 – Détections et extraction des embouts de câble RJ45 dans une scène encombrée avec une source de lumière complexe et des intrus jaunes

Sur le système ci-dessus (implémenté à l'aide d'OPENCV et C++), on parvient à extraire les trois embouts de câbles en temps réel sans extraire les autres items avec une forte similarité colorimétrique. Comme nous l'avons évoqués précédemment, pour toute scène, un nombre n d'embouts détecté(s) implique la présence de $n+n \pmod 2$ câbles ; nous avons donc dans cet exemple détectés deux câbles, ce qui correspond à la vérité sur le terrain. Les résultats obtenus sont très satisfaisants du fait que nous avons extraits les embouts partiellement occultés dans un espace complexe et encombré où la luminosité de la scène varie fortement (rayons de soleil) et avec une caméra de faible qualité (ancien système d'acquisition en 640x480 pixels). Cela ne veut pas dire que la reconnaissance sera continuellement idéale comme ici et on peut déjà prévoir l'apparence de faux positifs ou de faux négatifs (dans des cas qui, néanmoins, n'arriverons pas dans le cadre du projet). La solution que l'on proposera alors sera d'élargir la tolérance colorimétrique pour épandre le nombre de positifs extraits (quels qu'ils soient). Nous trierons enfin ces positifs en les filtrant ; cela peut être réalisé par matching (points d'intérêts) ou classification (SVM appliqué à la texture, par exemple). En effet, la base de l'embout du câble présente des rainures lorsqu'un post-it ou une carte propose souvent de grand aplats de couleurs sans texture (au mieux, des caractères variants). Ainsi, sur le cas de la figure 1, cette évolution extrairai les trois embouts, le post-it et le badge avant de post-valider par texture et écarter les deux dernières détections, qui sont des faux positifs.

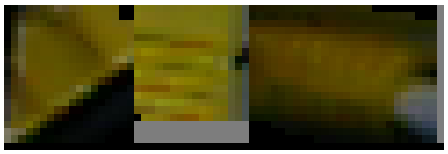


FIGURE 2 – Extractions réalisées dans notre exemple.

Avec cela, notre système reconnaissance axé sur la détection de câbles RJ45 aux caractéristiques connues et invariantes est réalisée. Elle est robuste à la plupart des opérations et ses seuls points faibles (dans notre contexte) est l'occultation complète et la non-conformité (embout remplacé ou atténué par une étiquette, un trait de marqueur etc.).

2.1.6.2 Câbles RJ11

Les câbles RJ11 sont plus difficiles à reconnaître que les câbles RJ45. Leurs embouts sont beaucoup plus petits et généralement d'une couleur moins éclatante. Cela signifie qu'un système d'acquisition minimal sera trop pauvre pour traiter le problème avec la même approche que pour les câbles RJ45. En effet, l'occultation est aisée et la détection par couleur considérablement moins précise. Enfin, dans notre contexte, la variabilité intra-classe est cette fois-ci non négligeable ; les embouts sont de couleurs et de formes différentes.

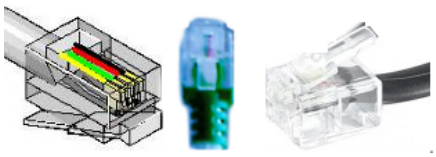


FIGURE 3 – Caractéristiques et variance des embouts de câbles RJ11

Approche structurelle :

Comment aborder ce problème, donc ? Nous commencerons par établir les caractéristiques communes de tous nos câbles RJ11 pour étudier celles ou les ensembles de celles qui sont les plus descriptives et stables aux yeux de notre système d'acquisition. Les propriétés du câble à retenir sont sa structure filaire et la présence d'une languette de forme invariante sur un embout variant. On peut déjà écarter la languette en tant que caractéristique en raison de la transparence du plastique utilisé (le signal acquis sera sensible à la transparence du matériel sans être capable de l'interpréter, on ne sait donc pas à quoi s'attendre). On ne peut travailler ni sur la forme, ni sur la couleur ou la texture des embouts car nous avons vus qu'ils varient trop (on risque alors d'introduire des détecteurs trop génériques sans être capable, derrière, de distinguer les faux-positifs des vrais). On aurait pu s'intéresser aux couleurs de fils sous le plastique transparent (voir la première représentation de la figure 3) car ceux-ci sont courts avec des couleurs faciles à isoler (vert et rouge). Cependant, la finesse de cette caractéristique et sa susceptibilité à être occultée sont trop importants (zone restreinte, visible uniquement sur une des faces de l'embout) en plus du fait que le choix de ces couleurs dépend du pays d'origine des câbles. Chercher à travailler avec des points

d'intérêts ne semble pas pertinent ; les embouts sont petits et cubiques, le système serait alors capable de reconnaître ceux-ci parmi un nombre envahissant d'autres détections dans la scène (qui, rappelons-le, est composée de plusieurs objets) et le tris serait alors impossible (car nous manquons de critères de filtrage). Il ne nous reste que la **structure filaire** du câble.

Dans le contexte du projet, les câbles peuvent être noués, soigneusement rangés ou dédaigneusement jetés dans leurs boîtes. On peut alors identifier plusieurs groupes de structures pour le corps des câbles :

FIGURE 4 – L'aspect filaire du câble

Dans un premier temps, les câbles peuvent être croisés et noués. Ce qui les caractérisent est alors l'entre-croisement de la structure : la section d'objets fin et long entre eux. Dans un second temps, le câble peut être attaché ou n'avoir jamais été utilisé, il est donc soigneusement enroulé. Ce qui va alors caractériser la structure sera la forme ovale ou circulaire de l'ensemble et la superposition de contours incurvés parallèles fins ; on peut interpréter cela comme une texture rayée. Enfin, le câble peut posséder un fil à "accordéon" ; ici aussi ce sera la texture qui va nous intéresser. Nous venons donc d'établir à notre sens les meilleures caractéristiques pour la détection par approche structurelle dans une scène désordonnée pour cette famille d'objets. Dès lors, **deux problèmes** nous apparaissent instinctivement : **dénombrer** avec précision les câbles semble ardu (car on s'intéresse à leurs structures et non pas à ce qui les délimites) et réaliser la **segmentation** de ceux-ci avec les cordon de blocs d'alimentation ou autre difficilement réalisable ; cependant, nous savons que des câbles sont présents. À défaut de pouvoir les classifier précisément, nous établirons néanmoins une façon de les localiser vulgairement pour attester de leurs présences.

La première approche pour la localisation consiste à détecter les régions répondant aux critères suivants : contours de câbles croisés, contours de structures de câbles parallèles. Les zones avec une densité suffisante de régions d'intérêts seront groupées et considérée comme en présence de câbles. Les cas isolés seront ignorés. Les détections de câbles partiellement superposées ou englobées seront considérées comme une seule détection. Ainsi, la méthode peut être illustrée de cette façon :

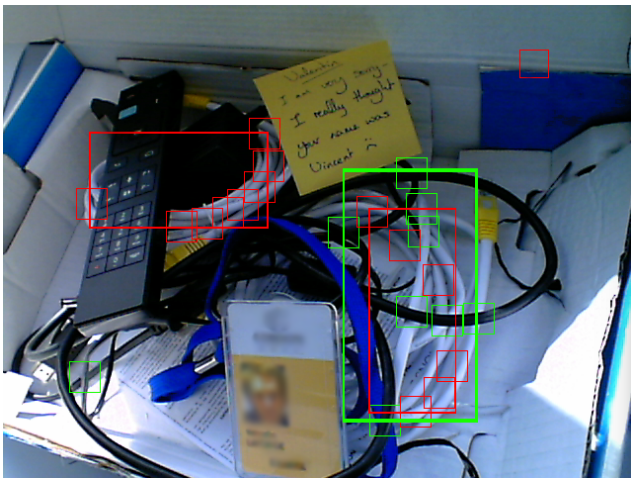


FIGURE 5 – Approche théorique et illustrée de la méthode.

Sur 5 ci-dessus, les carrés rouges détectent une texture filaire parallèle (on connaît le diamètre moyen des câbles et leurs distances avec la caméra) ; les carrés verts reconnaissent un croisement de câbles. On forme des groupes de régions d'intérêts, selon une tolérance définie à l'écart de distance entre deux ROI (mettons que deux ROI appartiennent au même groupe si moins de x pixels les séparent, selon la taille de la matrice) ; on localise le câble en traçant un rectangle englobant passant par les centres des carrés les plus éloignées du groupe. Si deux groupes se superposent en grande partie (sur plus de 70% de leurs surfaces, comme ici à droite), on choisit arbitrairement qu'il s'agit d'un même objet. Pour segmenter efficacement les zones tout en étant flexible à l'occultation, on résonne par "aimants". Sur la figure 5, le groupe rouge de gauche pourrait tolérer le groupe de droite car certaines régions sont relativement proches (avec un seuil de tolérance trop élevé, la méthode de localisation sur les ROI de textures rayées aurait détectée un seul très grand câble). Pour pallier à cela, on utilise une validation en croisant les méthodes. Si la distance entre deux points est tolérable mais qu'une région appartient géométriquement à un autre groupe, alors les deux ROI ne font pas parties du même groupe. Cette méthode de validation aurait permis de segmenter le groupe rouge dans le cas où la tolérance aurait été trop élevée.

Pour conclure la théorie, la méthode aurait ici détectée deux câbles. Les cordon courts des boîtiers d'alimentation ou des adaptateurs de prises téléphoniques auraient étés ignorés, ils ne sont pas suffisamment longs pour être à l'origine d'un groupe de ROI. Ils auraient, au mieux, provoqués des cas isolés ou un groupe faible. La méthode réagit donc mal au hasard ; lorsque tous les cordons que l'on ne souhaite pas détecter sont superposés, ils peuvent dans des cas très rares provoquer la création d'un nouveau groupe. Enfin, la méthode nous a uniquement permis de dire qu'il existe des câbles. La classification est donc générique.

Des expérimentations ont été menées pour tester la méthode. La première approche fut d'extraire les contours et de travailler sur la texture des bords en réalisant du Matching. On utilise alors différents pattern qui sont à reconnaître dans la scène.

FIGURE 6 – Exemples de pattern testés pour matcher des contours

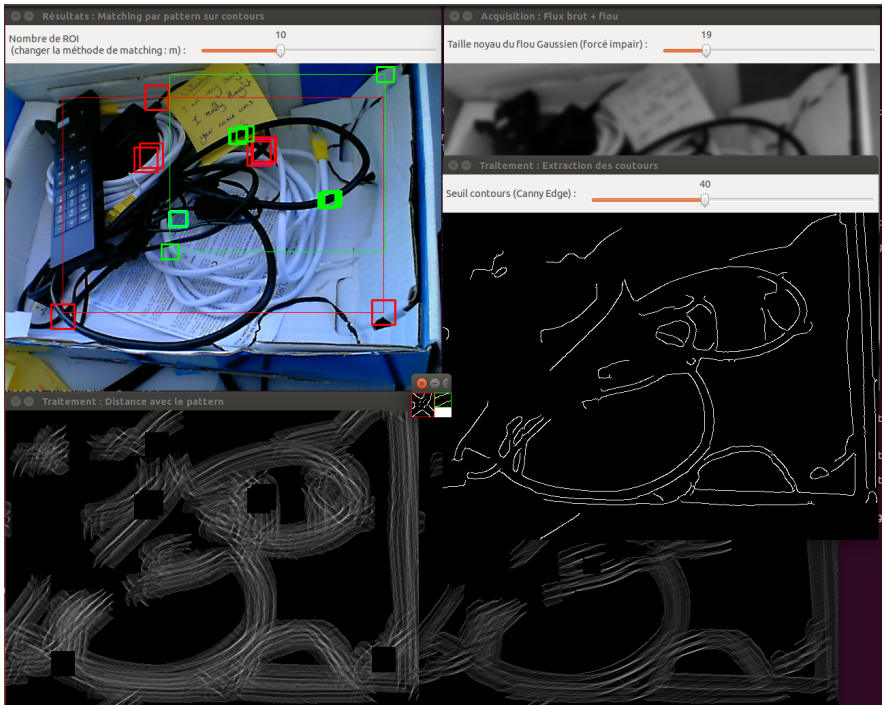


FIGURE 7 – Environnement implémenté pour la correspondance multiple de modèles de câbles croisés et parallèles.

Dans cette implémentation (Figure 7) sans segmentation des groupes, on réussi à extraire les zones caractéristiques de fils croisés et parallèles. Comme on pouvait s'y attendre, les textures de ces contours sont communes dans la scène et on détecte quelques faux-positifs (sur la Figure 43a, Annexe [3] , nous avons deux détections non souhaitées, mais isolées dans les coins droits). Selon la référence utilisée et sa complexité, on détecte parfois les bords d'objets quelconques. Segmenter les groupes, selon la méthode vu précédemment, aurait permis d'obtenir deux groupes au centre de l'image. Nous avons donc des résultats insuffisants mais nous pouvons toutefois chercher à améliorer la reconnaissance en combinant notre méthode avec d'autres approches d'interpolation des structures de l'objet étudié.

Combinaison d'approches structurelles :

Le système de reconnaissance de câbles proposé est fonctionnel dans un cadre connu, où la scène et les objets qui la compose sont à une distance connue. Les images de références utilisées pour la correspondance ne sont robustes ni à la rotation, ni au changement d'échelle (d'où

l'utilisation de formes de "croix" génériques). Enfin, la méthode détecte des faux positifs. Pour combler ces problèmes, nous pouvons éventuellement combiner cette approche avec les suivantes :

- **Spline** [20] / **courbe Béziers**[19] Après extraction des contours sur notre scène, nous cherchons à réaliser des interpolations polynomiales sur les contours incurvés. Nous partons du principe que seuls les câbles possèdent une structure courbée et nous tirerons avantage de cette propriété pour établir que toute bordure arquée indique la présence d'un câble.
- **Propagation de labels**[8, 16, 13] Nous propageons des labels sensibles à nos contours depuis le centre des ROI détectées (le centre des pattern de références doivent donc être des corps de câbles). Ainsi nous obtenons une segmentation colorée de la scène et nous approchons cette segmentation sur la forme. Nous essayons alors de faire correspondre des ovales (pour les câbles attachés) et des formes en buissons (pour les câbles emmêlés). En nous séparant de toutes les régions cubiques de la segmentation, nous espérons alors isoler les câbles.
- **Transformée de Hough Généralisée** En 1962, P. Hough propose une méthode de reconnaissance de lignes pour les images numériques [14], en 1972 D.H. Ballard généralise la méthode [9] proposée par Hough pour la reconnaissance de formes arbitraires [4]. Ces méthodes peuvent nous permettre d'affiner notre système de reconnaissance de formes de deux façons. La détection de lignes peut être utilisée pour détecter les objets cubiques de l'image. La transformée généralisée peut nous permettre de reconnaître les formes plus complexes tels que les arcs de cercles. Ainsi, dans le premier cas, nous reconnaissons les zones à ne pas traiter et dans le second, nous confirmons la présence de câbles.
- **Estimation du squelette de la scène** : Une approche plus morphologique du problème consiste à estimer les squelettes [26] de la scène depuis ses contours. Si la méthode pousse la reconnaissance du câble au niveau d'établir sa position (emmêlé, croisé, attaché, plié…) dans une scène simple, elle répond mal aux scènes complexes, les armatures obtenues peinant à décrire un objet particulier.

Ces approches étant basées contours, nous devons approfondir notre approche de détection des contours des câbles. La méthode de détection de contours utilisée jusqu'à maintenant était celle de John F. Canny, proposée en 1986 et connue sous le nom de "***Canny Edge Detector***"[6]. Nous utilisions, auparavant, un flou gaussien au noyau interactivement ajustable pour effacer les détails précis (caractères, petit logos) dont nous ne souhaitions pas détecter les contours. En conservant cette dégradation, nous étudions maintenant la possibilité seuiller et combiner plusieurs méthodes d'extraction de contours pour avoir des pixels qui s'attachent davantage aux câbles et moins aux objets indésirables.
Une simple **binarisation seuillée** peut, dans ce cas, nous être utile. Depuis une image I, nous construisons une nouvelle matrice B à un canal d'intensité obéissant au filtrage suivante :

$$\forall I_{i,j} \begin{cases} si & I_{i,j} > S, \quad B_{i,j} = 1 \\ sinon & B_{i,j} = 0 \end{cases} \tag{1}$$

Ce qui nous intéresse dans cette binarisation basée intensité, c'est qu'elle possède l'inconvénient d'être à elle seule trop sensible à la lumière. Celle-ci met facilement les ombres portées

en évidence, entre autres. Nous l'utiliserons donc pour limiter les effets indésirables qu'un éclairage génère et pour appuyer la détection de contours dans le cas ou câble est de la même couleur que le fond (dans notre cas, blanc sur blanc).

Nous utiliserons aussi les résultats qu'un Laplacien porté sur deux opérateurs Sobel (un vertical, un horizontal) peut nous offrir. Celui-ci serra aussi calculé sur un flou Gaussien pour les mêmes raisons que vues précédemment. Ainsi, avec une combinaison des différents traitements sensibles aux contours, nous pouvons approcher au mieux les formes complexes de nos câbles.

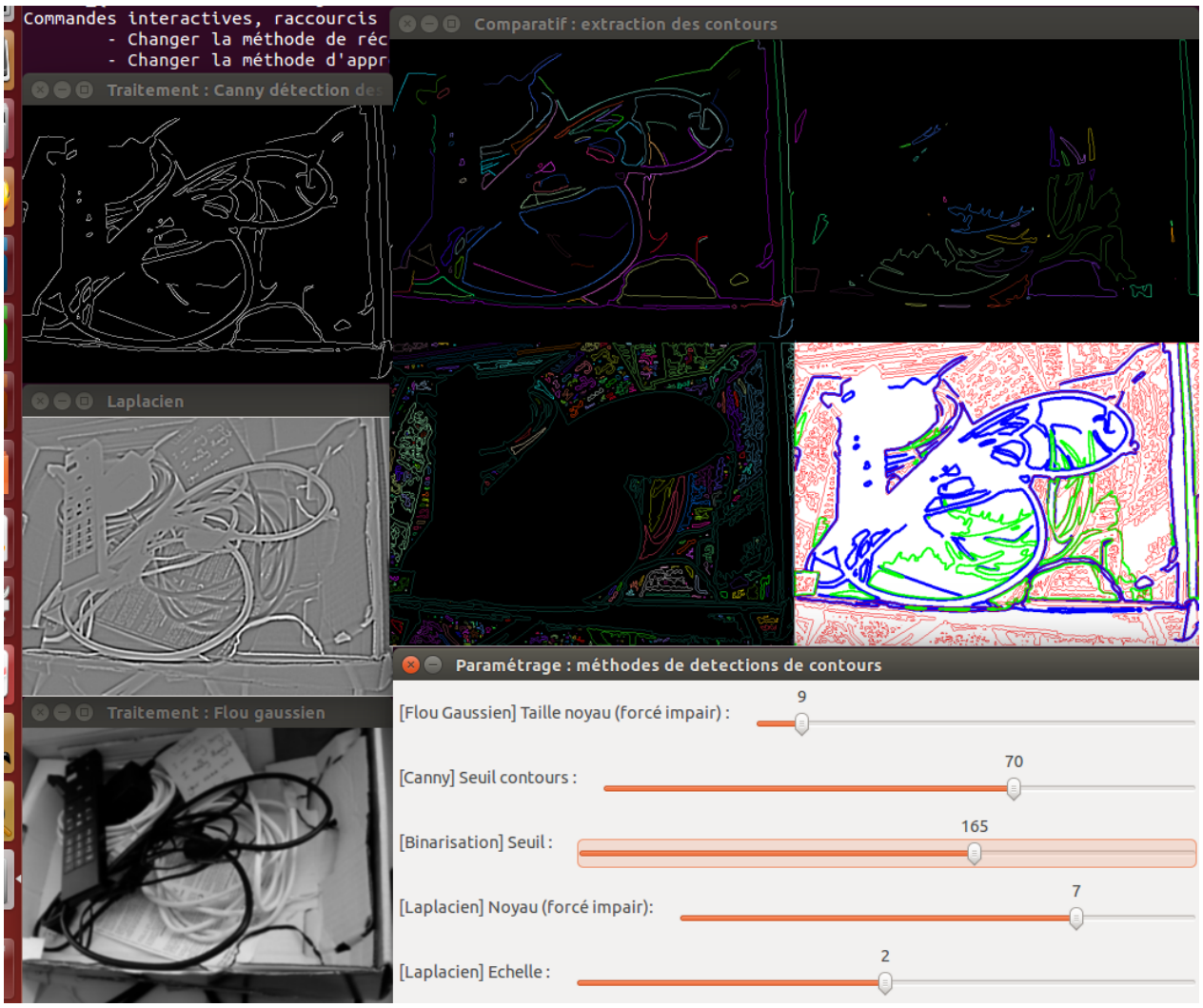


FIGURE 8 – Combinaison de méthodes d'extraction de contours

Enfin, les régions en présence de courbes dans notre matrice pourraient être extraites grâce à un détecteur spécialisé [7].

Approche polynomial :

Notre dernière approche pour détecter nos câbles RJ11 consiste à calculer les **orientations locales par blocs**. Cette méthode, utilisée entre autre en biométrie pour la reconnaissance d'empreintes digitales, nous permet de suivre les courbes d'une texture. Nous verrons qu'estimer les orientations locales pousse, dans le cadre de RECOB, la reconnaissance de contours à la segmentation de ceux-ci de façons relationnel. En partant du principe qu'une orientation pointant sur une autre indique une liaison entre les deux, on s'intéresse aux chaînes d'orientations incurvée pour détecter nos câbles.

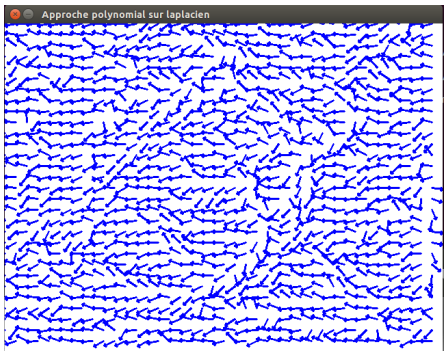


FIGURE 9 – Estimation des orientations locales par blocs de 16 pixels depuis le laplacien de notre scène floutée (Gauss)

Sur la Figure 9 ci-dessus (correspondant toujours à la scène, on remarque que les orientations incurvées suivent notre câble quand celles de la télécommande divergent entre elles. Les orientations horizontales marquent les zones qui ne nous intéressent pas. Avec ces données, nous pouvons lier les orientations locales qui se réfèrent entre elles pour créer des courbes, extraire un polynôme la décrivant et l'interpréter en tant que courbe ou droite.

En combinant les méthodes précédemment présentées, nous pouvons finalement :

1. Travailler sur l'image acquise en niveau de gris (la couleur n'aidera pas la détection),
2. Appliquer un flou gaussien pour lisser les détails superflus tels que les textes ou même les parasites,
3. Extraire les contours à l'aide des méthodes de Canny et de Laplace,
4. Seuiller et ouvrir morphologiquement le Laplacien obtenu,
5. Approcher les ombres portées à l'aide d'un opérateur binaire simple,
6. Soustraire les ombres portées à notre ouverture morphologique de laplacien binarisé,
7. Soustraire les contours non incurvés grâce à la transformée de Hough (bord de la boite),
8. Calculer les orientations locales par blocs,
9. Combiner les résultats.

Nous obtenons alors la détection de structure de câbles suivante :

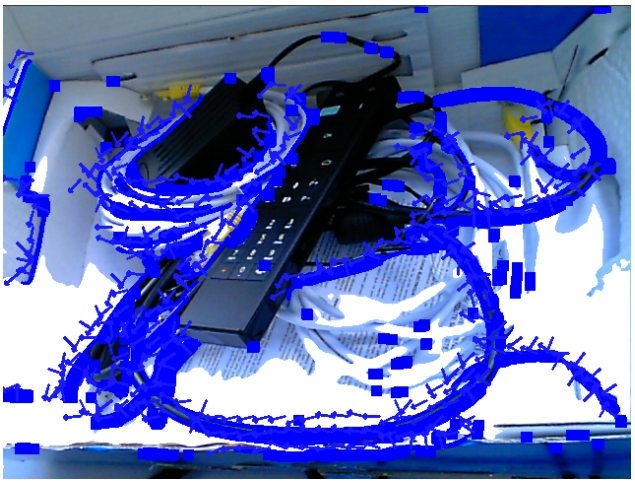


FIGURE 10 – Combinaison algorithmique pour la reconnaissance de structures courbées.

On peut dès lors estimer une magnitude à nos orientations locales et effectuer une opération de validation par pointage dont nous avons parlé précédemment :

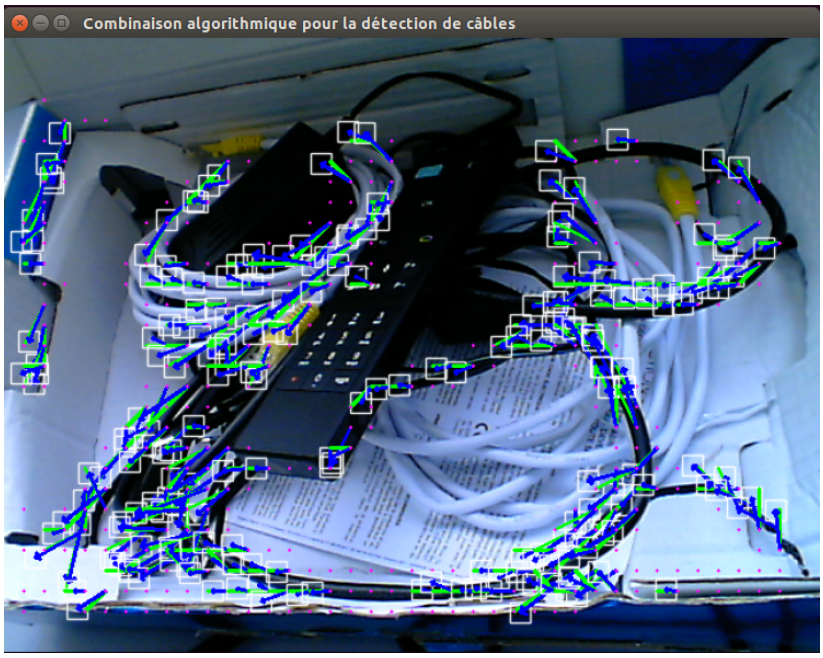


FIGURE 11 – On réalise les orientations locales avec poids sur les câbles, on valide les vecteurs qui sont référencés par d'autres (conservation d'une suite logique de points) puis on les relies. Plus tard, on éliminera les bords de carton avec la méthode de Hough (détection de lignes) puis on essaiera de réaliser une interpolation des points de vecteurs (trouver une courbe).

Nos résultats, visibles sur la figure 11, parviennent à suivre fidèlement nos câbles et sont en ce point satisfaisant ; toutefois, d'un point de vue critique et évolutif, la méthode possède quelque sérieux défauts. Premièrement, nos efforts sur le travail des gradients et contours sont à double tranchants. En effet, on constate au centre de l'image qu'une région

entière de câbles blancs n'est pas détectée. Les raisons derrière ce phénomène est que la tolérance à l'intensité des contours qui nous intéressent est bien trop stricte et que la variation du gradient des structures de câbles positivement détectées est amélioré par le voisinage (variation remarquable au niveau de l'histogramme des intensités). Ainsi, nous n'avons pas procédés à l'étape suivante qui auraient été d'extraire des polynômes de nos suites de vecteurs en relation pour les étudier.

Avec une cross validation de la méthode par pattern que nous avions précédemment établie, nous pourrions éliminer plus facilement les nouveaux cas isolés (qui sont les bords de cartons interrompus) et chercher à dénombrer le nombre de nos câbles. Encore une fois, nous sommes capables de détecter la présence de câble sans pour autant les classifier.

Étude d'une approche de reconnaissance cognitive inspirée du système biologique :

Si on confronte l'homme aux scènes étudiées précédemment, il sera capable de dénombrer trois, peut être quatre câbles en fonction du temps dont il dispose pour répondre à la question. Il est alors intéressant de se demander comment il peut les classifier et comment il procède à la segmentation des éléments.

Pour la question, **combien existe-t-il de câbles**, l'œil va parcourir l'image 'z' ou en 'x' sur ses points chauds, selon leurs attractivités définies par la densité, le contraste et la vivacité de la région. On comptera alors les objets en se basant sur les différences des caractéristiques des ensembles positifs et l'estimation de leurs contours dans l'espace. Si la scène est encombrée et les objets en grande partie occultés, on finira par rechercher des compléments d'informations (des indices) sur les régions très détaillées, très denses.

Pour la question, **comment segmente-t-il les câbles**, il va les différencier par propriétés (couleurs, largeurs) et estimations : Il réalisera une approche probabiliste de ce qu'il perçoit, argumentant qu'il est très probable que tel câble occupe tel espace et que tel section de câble appartienne à tel groupe de parties de câbles. L'homme est capable de cette performance car il a connaissance de la structure générique d'un câble et de l'occupation qu'il peut avoir dans l'espace ; il est capable d'interpréter les relations suscitées par les interactions des objets. Il détecte un câble grâce à sa connaissance de l'objet en question ainsi que des objets en relations et est capable de construire une image mentale de sa continuité logique sur les parties occultées en fonction de la profondeur qu'il perçoit. Lorsqu'il se trompe, c'est que l'information sur un objet n'est pas suffisante, qu'il découvre un ensemble nouveau à sa connaissance ou que sa reconstruction est imparfaite ; il témoigne dans ce dernier cas d'un effet d'optique ou de surprise. Reproduire un système de vision similaire à l'homme requiert donc de posséder une base de connaissance (acquise par apprentissage), d'être capable de comparaison (matching) ou de sensibilité à certaines propriétés (détection de couleurs, contours et textures) et d'interpréter la profondeur de la scène afin de comprendre l'organisation des relations entre les éléments (superposition, équilibre). Une nouvelle approche de notre problématique de reconnaissance de câbles pouvant être portée au cas général serait donc d'un côté, d'inclure un système d'apprentissage de formes basiques sensible à la flexibilité de la structure et de l'autre, d'interpréter la profondeur afin de réaliser une approche probabiliste par calques

(fond de scène + x calques de superposition(s) + premier plan). Ainsi nous cherchons à reconstruire autant de représentations matricielles de notre scène qu'il existe de superpositions. Enfin, par différentes méthodes (Inpainting, propagation) nous pouvons essayer de reconstruire les zones sans informations sur les calques ainsi créées. La reconnaissance serait alors effectuée sur chacun des calques de l'image. Un tel système ne posséderait pas, tel que décrit, de la capacité d'adaptation de l'homme mais présenterais, en contre-partie, une capacité de reconnaissance plus rapide et moins sensible à la fatigue.

2.1.6.3 Télécommande

La télécommande est un cas très particulier dont la forme et la surface sont très caractéristiques. On peut effectuer la reconnaissance d'un tel objet au niveau de ses boutons.

Caractéristiques locales :
Cette méthode offre de bons résultats pour un objet susceptible de lever beaucoup de descripteurs (coins et surface des boutons). Une solution possible est l'utilisation d'algorithmes de détection et de description de point d'intérêts locaux comme décrits dans l'état de l'art (voir Annexe [1]).

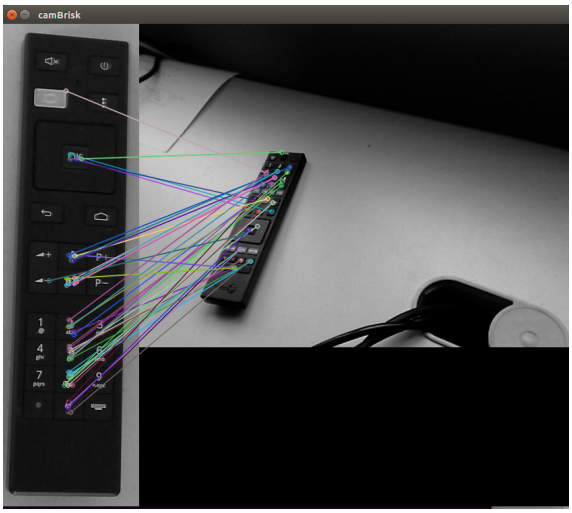


FIGURE 12 – Approche expérimentale de reconnaissance avec BRISK. On utilise ici une référence, une échelle, une orientation, un angle différents de l'objet sur la scène.

L'image ci-dessus démontre un début d'implémentation encore mal paramétré de la méthode. Le développement de cette solution a avortée en faveur d'autre approches que nous verrons dans la prochaine partie.

2.1.6.4 Réseaux de neurones

Expérimentations :

A des fins expérimentales, nous avons approché la détection de câble en utilisant un algorithme d'apprentissage profond et convolutif : le réseau de neurones de GOOGLE, nommé

Inception-v3 [23]. Celui-ci peut déterminer la classe d'un élément présent sur une image parmi 1000 classes avec une précision n'admettant que 21.2% d'erreurs sur sa meilleur estimation (top-1) et 5.6% sur l'ensemble de ses 5 meilleurs (top-5). Étant conçu pour être utilisé sur des images fixes, son coût de calcul est de 5 milliard multiply-adds par inférence et, au moment de la rédaction de ce rapport, demande moins de paramètres que les propositions de Deep Learning concurrentes (moins de 25 millions). Il est possible d'adapter Inception-v3 à des tâches de reconnaissances spécifiques (à notre problématique, par exemple) en ré-entraînant uniquement la dernière couche du réseau lors d'une nouvelle phase d'apprentissage. Celle-ci est très coûteuse à réaliser : les systèmes utilisés pour ce genre d'opérations, souvent équipés de matériels approchant les 128GB de mémoire morte connectées à 8 cartes graphiques NVIDIA Tesla K40 requièrent de longs jours ou semaines pour la finaliser (selon la base d'apprentissage, des actions supplémentaires de prétraitement etc.). Nous limitons donc les coûts d'apprentissage grâce au modèle pré-entraîné qui est mis à disposition des chercheurs (`http://googleresearch.blogspot.fr/2016/03/train-your-own-image-classifier-with.html`), nous procédons donc par Transfer Learning. Ainsi, nous avons pu ré-entraîner le réseau sur nos propres catégories et étudier les résultats de la détection sur nos objets. Compte tenu de la faible capacité calcul et de mémoire de notre carte graphique (Quadro FX 580 avec 512mb), nous utiliserons dans un premier temps le CPU (8 cœurs, 2.4GHz) et la mémoire vive (16 Gb) de notre ordinateur. Les craintes étaient alors les suivantes :

1. Apprentissage de formes dans un état particulier pour les câbles (enroulé, emmêlés, autre),
2. Segmentation des câbles et des boîtiers à extension filaire difficile,
3. Impossibilité du dénombrement ou de classification du type de câble.

Celles-ci seront plus ou moins matérialisées au cours des expérimentations, mais nous verrons que sans être négligeables, elles n'empêchent pas la reconnaissance d'avoir lieue.

Dans un premier temps, nous avons testé la fiabilité de l'apprentissage sur notre machine depuis une base de 3175 images connues et déjà étudiées (5 types de fleurs). Après un certain temps, 15000 pas ont été réalisées et nous stoppons l'apprentissage pour évaluation :

```
Succesfully loaded model from /home/valentin/Documents/RECOB/RDN/flowers-train/model.ckpt-15000 at
     step=15000.
2016-05-10 09 :43 :31.639501 : starting evaluation on (validation).
2016-05-10 09 :47 :06.322086 : precision @ 1 = 0.9590 recall @ 5 = 1.0000 [512 examples]
```

Cette précision de 95,9% a été atteinte en 5 jours, évaluée en 5 minutes et calculée sur un lot aléatoire d'images composant la base (ce qui n'est pas une bonne pratique et ce que l'on évitera lorsque nous travaillerons sur notre problématique). Compte tenu de la similarité des fleurs cela est suffisamment satisfaisant, mais nous garderons à l'esprit que des entraînements plus poussés mènent à 99,7% de précision pour cette base d'entraînement. On peut, dans notre cas, être moins exigeant et entraîner cette base sur 4000 pas. Nous obtenons alors 91.8% de précision finale pour environ 4 à 5 heures d'entraînement. Il est intéressant d'observer que la précision ne s'améliore pas de façons linéaire avec l'amplitude de l'entraînement (voir 1 en Annexe [4]). En réalité, elle peut même être amenée à baisser d'un pas à un autre dû à la présence d'un générateur aléatoire dans le réseau ; cependant, elle convergera toujours vers une précision optimale. Les expériences ont montré qu'entraîner consécutivement le même

réseau dans les mêmes conditions et sur la même tâche semble l'affaiblir (87,9% ; 86,9% ; 85,6% ; 84,6% puis 84,8%). L'attente générée par l'entraînement et l'évaluation des résultats nous as permis de constituer (voir partie 4.2.1.1 pour les bonnes pratiques) une base d'image pour notre problématique de reconnaissances d'objets :

A) 160 rj11

B) 29 chargeur

C) 109 télécommandes

D) 50 adaptateur

Celle-ci est composée d'images récupérées sur internet et d'acquisitions obtenues rapidement grâce à un bout de code écrit sous OpenCV tirant partie d'une webcam. Un pré-traitement a été effectué pour correspondre aux exigences décries dans la partie 4.2.1.1. Nous avons connaissance du manque numérique d'image pour chaque classe (la tâche consume énormément de temps, même avec nos scripts de formatage de la base). La reconnaissance est ici calculée selon ces 4 classes, par probabilité et sur des scènes beaucoup plus complexe que les références. Il convient donc d'interpréter correctement ces résultats :

— 80% - x% - y% - z% , il y a un item bien identifié
— 40% - 40% - x% - y%, il y a deux items bien identifiés
— 33% - 33% - 33% - x% , 3 items bien identifiés
— 25% - 25% - 25% - 25%, tous les items sont présents

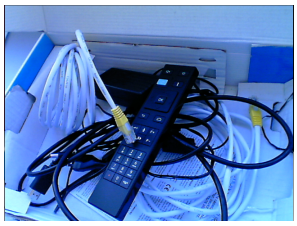
Nous n'évaluerons pas le réseau sur des scènes complexes sans présence d'au moins un de ces objets, car l'interprétation ne tiendrais plus (et cela ne correspondrait pas au bien du projet RECOB). En effet, le cas échéant, le réseau forcerai la perception pour faire correspondre ces 4 classes à la scène : il trouvera toujours un item avec une présence supérieure ou égale à 25% de précision. De même, si un objet est plus visible qu'un autre, son pourcentage de présence sera plus grand que le second. Nous utilisons des cas absurdes pertinents (texture ou forme trompeuse) par curiosité scientifique et pour étude du réseau entraîné. Nous effectuons donc les évaluations sur trois réseaux, en faisant varier la taille de la base d'images et la durée de l'entraînement.

A) RDN-C 1 : Faible base d'images à 3 classes (RJ11, Chargeur et Télécommande)

B) RDN-C 2 : Entraînement à 4000 pas sur base à 4 classes (ajout de l'adaptateur pour complexifier la détection de câbles ; précision : 91,6%)

C) RDN-C 3 : Entraînement à 128000 pas sur base à 4 classes (précision : 86,0% ; ici nous aurions dû récupérer un nouveau réseau pré-entraîné afin d'éviter les conséquences d'un réseau entraîné consécutivement sur le même jeu de données).

Ceux-ci se verront attribués la tâche de reconnaissance sur les 7 scènes suivantes :



(a) Scène 1

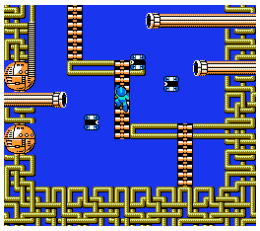
(b) Scène 2

(c) Scène 3

(d) Scène 4

(e) Objet de type filaire sous forme de boîte

(f) Objet de type boîte avec texture filaire



(g) Scène complexe avec structure similaire aux objets étudiés

FIGURE 13 – Scènes évaluées par le réseau de neurones

TABLE 1 – Comparatifs des détections d'objets ciblés par réseau et par scène

Scène	Round	RJ11	Adaptateur	Chargeur	Télécommande
1	1	**87**	-	12.8	0.1
	2	**73.22**	11.8	13.42	15.7
	3	**93.6**	3.01	2.9	0.48
2	1	**78**	-	10.49	10.8
	2	**80.12**	5.68	0.8	**80.12**
	3	4.93	1.25	0.37	**93.78**
3	1	**81.85**	-	18.09	0.05
	2	**62.56**	18.84	18.51	0.08
	3	**72.58**	18.07	9.33	0.009
4	1	**84.02**	-	13	2.2
	2	**73.45**	6.92	13.22	6.45
	3	**93.91**	0.82	2.75	2.52
Corde	1	**90.8**	-	2.11	7.05
	2	**42.77**	**34.73**	3.15	19.34
	3	**37.57**	**47.16**	0.9	14.29
Chocolat	1	19.8	-	1.5	**78.69**
	2	12.67	5.44	1.61	**80.28**
	3	5.97	0.79	0.24	**92.99**
complexe	1	22.75	-	**35.75**	**41.49**
	2	11.65	8.63	24.15	**55.83**
	3	3.81	4.02	18.60	**73.55**

Critique : Pour commencer, le réseau pourrai être mieux entraîné. On devrait ajouter beaucoup plus d'images dans notre banque et, puisque nous n'avons pas les ressources pour

créer une large base d'images, compenser en réalisant quelques distorsions (surtout sur les luminosités, couleurs, même si ce n'est qu'une compensation et une mauvaise pratique si l'on dispose de suffisamment d'images). Des acquisitions avec une autre caméra auraient aussi pu aider à diversifier les bases d'images. On remarque que la classe RJ11 prédomine fortement car le réseau est entraîné à reconnaître une architecture filaire, aussi présente sur l'adaptateur et le chargeur. Certaines non détections peuvent sembler curieuses aux yeux de l'homme ; il s'agit, pour la plupart, de prédominance par la classe la mieux entraînée : les câbles RJ11. Il est donc intéressant d'observer les autres résultats admis (Top-3) quand la classe RJ11 arrive en Top-1 erreur. Enfin, en vue de ces résultats nous pouvons prédire que le réseau ne permettra pas de distinguer les classes RJ11 et RJ45. L'idée de cette reconnaissance intra-classe par cette approche semble néanmoins concevable si ce dernier s'entraîne sur des bases d'images ridiculement colossales (plusieurs milliers pour chaque type de câble). Une autre approche à avoir consiste à constater les différences colorimétriques de ces deux sous-classes et de nourrir le réseau en grande partie sur ces aspects spécifiques. La connaissance serait alors constituée de nos câbles spécifiques (Objet → Câbles → Câbles RJ11 → Câbles RJ11 Spécifique) et la reconnaissance pourrais être troublée par la couleur des fils des chargeurs et adaptateurs qui seraient dès lors davantage évalués sur leurs couleurs.

On s'intéresse maintenant à la capacité du réseau pour reconnaître des objets segmentés, avec une différence intra-classe mais appartenant toujours à la même catégorie. Par exemple, nous évaluons des télécommandes différentes de celles qui nous ont permis d'entraîner le réseau. Nous réalisons alors grossièrement la moyenne de quatre résultats obtenus sur des images aléatoires :

A) Télécommande : environ 99,9% (99,96% ; 99,86% ; 99,98% ; 99,76%)

B) RJ11 : environ 91,1% (99,98% ; 75,38% ; 94,58% ; 98,45%)

C) Adaptateur téléphonique : environ 91,8% (86,75% ; 84,13% ; 98,24% ; 98,01%)

D) Chargeur : environ 66,53% (81,63% ; 3,37% ; 93,01% ; 88,13%). Sur notre deuxième test, le réseau a échoué à différencier le chargeur d'un câble RJ11 et peine généralement davantage à dépasser les 90% de précision. La raison derrière ce phénomène est que tout deux possèdent des structures filaires très similaires. Curieusement, la pire reconnaissance (3,37%) a lieu sur le chargeur le plus similaire à nos références aux yeux de l'homme. En vue de l'image utilisée, nous pouvons en déduire que le réseau de neurones est plus sensible à l'agencement des objets que le système de reconnaissance de l'homme, qui est capable d'interpréter la profondeur et l'occultation.

Les résultats obtenus étant déjà corrects, nous pouvons désormais chercher à enrichir notre entraînement. Cela est réalisable en nourrissant davantage notre base d'image ou en appliquant des mutateurs à nos images existantes. Muter les images consiste à effectuer des opérations de modification sur celles-ci. En effet, en appliquant des modifications de luminosité à la volée sur les entrées du réseau de neurones, nous espérons rendre notre reconnaissance plus robuste. Le contrecoup de l'opération est le temps requis pour affecter la modification, qui rend l'entraînement 120x plus long (grossièrement) estimé.

***Note** : Malheureusement, après un mois et à 24h de la fin de l'entraînement, l'ordinateur utilisé a crasher inopinément et l'expérience a fatalement faillie. Pendant ce temps, d'autres*

tâches nous ont été affectées et l'expérience utilisant des distorsions d'images est passée au second plan pour finalement ne pas être réalisée. Enfin la journée Deep Learning réalisée plus tard à l'université de Caen nous confirmera que les résultats n'auraient étés que très peu meilleurs.

Nous avons, par ailleurs, tenté de contrecarrer l'aspect négatif d'entraînements consécutifs par un entraînement 32x plus long (128 000 pas). Si la précision a augmentée (de 2% environ) par rapport à la baisse engendrée par le manque de diversité lors des entraînements consécutifs du réseau, cela ne suffit toujours pas à atteindre la précision que nous avons eue lors du premier entraînement. La plus-value du Transfer Learning est donc ici, encore une fois démontrée. La meilleure façon d'affiner un réseau pré-entraîné est donc, d'après nos expériences :

- d'utiliser un réseau paramétré sur de larges bases d'images et des classes génériques de préférence différente des nôtres (*privilégier la diversité*),
- d'entraîner suffisamment longtemps le réseau pour le laisser converger vers sa précision de reconnaissance optimale (*apprendre suffisamment*),
- de travailler sur une base d'images bien conçue (*apprendre ce qu'il faut)*

Segmentation intelligente :

Comme nous avons pu le découvrir dans la partie 2.1.6.2, le cerveau humain ne fait pas que de l'estimation de similarité dans le processus de reconnaissance d'objets. Celui-ci **traite au préalable** le signal visuel qu'il reçoit et reconstitue ce qui lui échappe par occultation. Il possède donc cet avantage sur nos réseaux de neurones qui n'effectuent pas de traitement des entrées. Nous allons ici approcher le problème de reconnaissance en s'inspirant un peu plus du fonctionnement supposé du cerveau biologique et de ses capteurs. Notre objectif est cette fois de **simuler en partie la capacité de groupage du cerveau** en segmentant les objets d'une scène. Chaque segment sera alors fourni à un réseau de neurones convolutif pour reconnaissance. En accord avec nos expérimentations sur le comportement des réseaux de neurones (voir partie 2.1.6.4), on peut s'attendre à de très bon résultat avec cette méthode (suivant la qualité de la segmentation).

Pour segmenter les objets d'une scène complexe, il faut grouper et savoir reconnaître les contours d'objets. Jusqu'ici, nous utilisions le gradient de nos acquisitions 2D pour cette tâche. Néanmoins, ce type de détection ne permet pas toujours de reconnaître les contours créés par superposition d'objets (câbles par exemple). Ainsi, sur un plan à deux dimensions, on détecte des contours finis d'éléments assemblés qui ne sont pas des objets. Pour renforcer la détection de contours d'objets, nous pouvons comme l'homme, utiliser la **profondeur**.

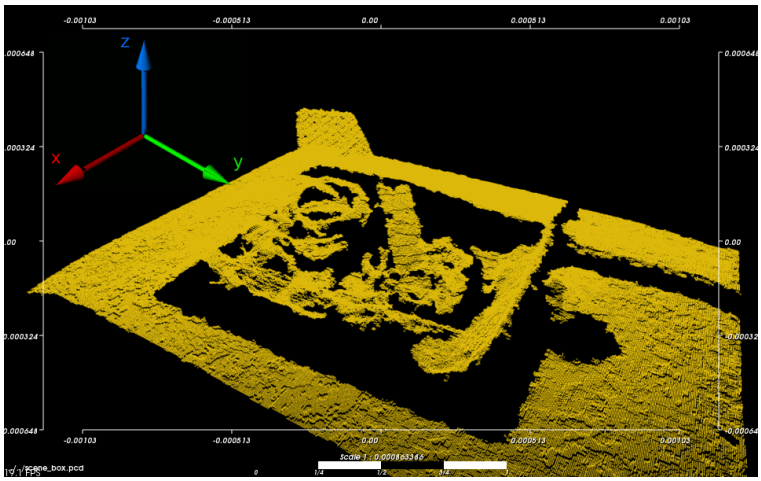


FIGURE 14 – Nous travaillons désormais sur la profondeur (Z)

En travaillant sur un plan à trois dimensions, nous sommes alors capables de détecter l'occultation et d'ignorer les contours de gradient issus de textures. Ainsi, nous segmenterons la scène par plans (ou calques). **La première approche est alors d'effectuer un filtre par profondeur** :

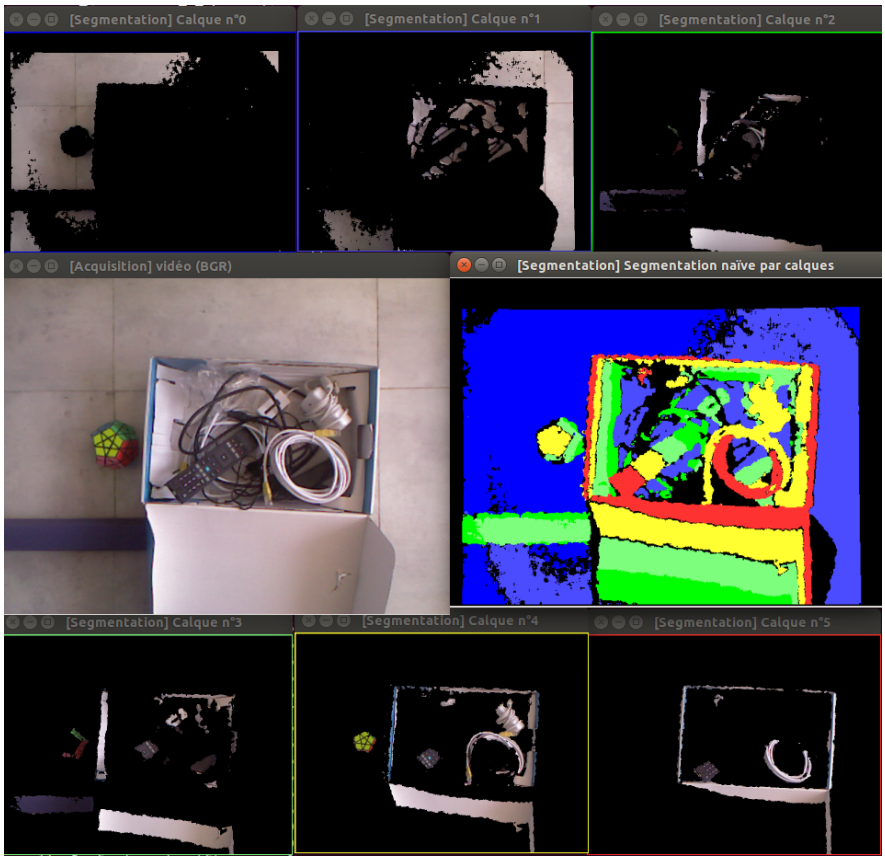


FIGURE 15 – Segmentation naïve par 6 calques de profondeur répartis sur la hauteur de la boite

On peut sans grande surprise observer que cela est **insuffisant**. La segmentation a lieu sur un plan XY quand les objets varient sur XYZ ; ce manque de flexibilité est mis en évidence sur la télécommande dont une extrémité repose plus en hauteur que l'autre : elle est alors segmentée sur 4 calques. Nous souhaitons donc pouvoir **reconnaître des plans capables de se croiser**. Nous effectuerons donc une détection planaire répondant à une problématique de groupements de points (sous forme de clusters) d'un nuage représentant la scène. La méthode est alors la suivante :

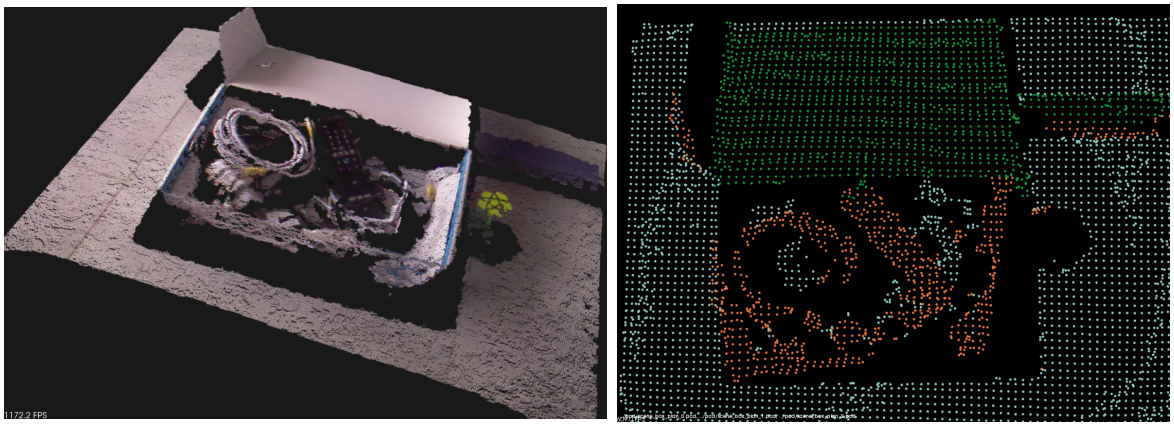
A) Acquisition de la scène par un capteur de profondeur,

B) Échantillonnage de la scène (principalement pour des soucis d'optimisation, mais aussi pour rendre plus abruptes les variations sur XYZ),

C) Détection de formes planaires en utilisant la méthode de RANSAC,

D) Segmentation d'images RGBD suivant les plans détectés.

E) Passage au réseau de neurones des plans détectés.



(a) Visualisation (b) Segmentation

FIGURE 16 – Segmentation planaire

Sur l'image ci-dessus, nous avons réussi à segmenter 3 plans sur un peu moins de 10cm de profondeur. Nous avons donc en bleu, le fond (ici, le sol), en vert le couvercle de la boîte et en orange les éléments contenus dans cette boîte. L'approche est donc meilleure que la précédente car la segmentation de la télécommande est complète. Nous pouvons alors chercher à reconstruire le plan qui nous intéresse :

FIGURE 17 – Reconstruction de la segmentation planaire de nos objets

Sur la figure 17, les trois premières fenêtres répondent à des soucis de calibrage des données des capteurs (notre capteur est peut-être trop près des limites établies pour l'acquisition optimal (voir 4.1.6). L'approche est meilleure que la précédente. Elle nous permet de segmenter en particulier les objets consistants dans la scène (télécommande, adaptateur téléphonique, chargeur).

Amélioration et objectifs sous-jacents de nos approches :

Si nous nous sommes précédemment intéressés au système cognitif humain et si nous avons amélioré progressivement les méthodes de segmentations, c'est afin de nous rapprocher d'une méthode de segmentation intelligente permettant d'enrichir significativement l'étape suivante de reconnaissance par RDN qui suivrait. Que ce soit par manque d'expérience ou par manque de temps, la méthode vers laquelle nous nous dirigions n'a pas pu être implémentée et ne le sera pas ; toutefois, parce que nous estimons qu'elle mènerait à de bons résultats dans notre cas (scène complexe d'objets désorganisés et s'occultant les uns les autres), nous prendrons tout de même quelques instants pour la décrire.

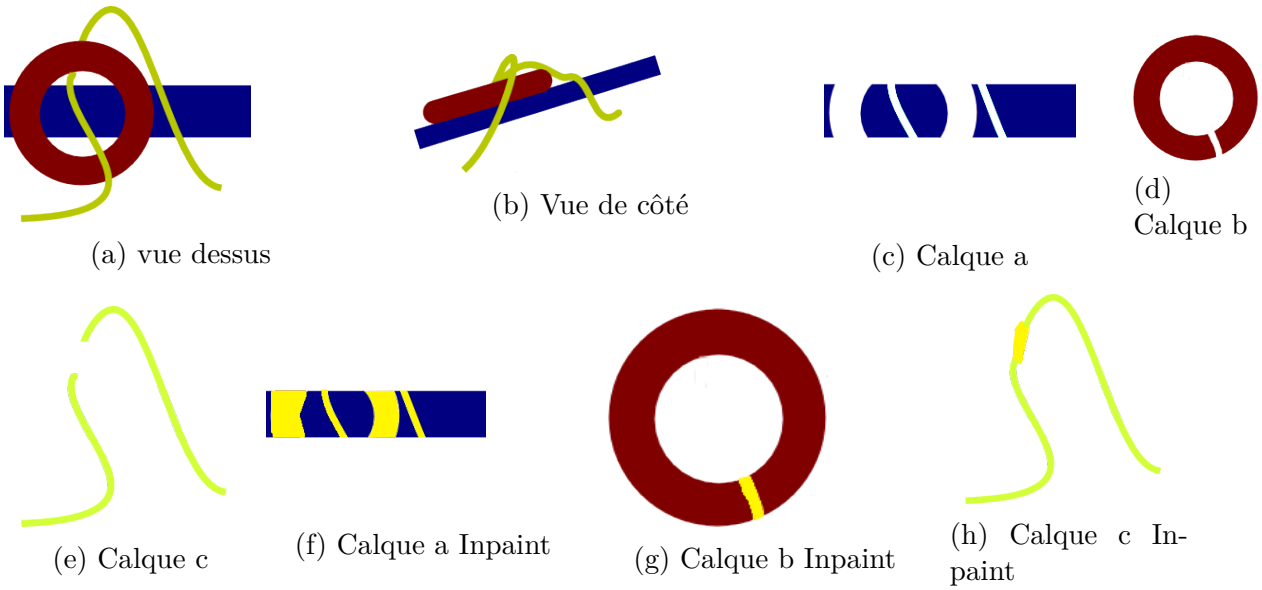
Depuis une caméra et un capteur de profondeur, la méthode pensée viserait à isoler des objets sur des calques en réalisant un aplat de leurs textures, selon un angle d'acquisition constant. Nous traiterions alors 6 canaux par pixels (XYZRGB). Sur la figure 18 nous décri-

vons cette méthode de façon simple et schématique.
Dans un espace en trois dimensions où X est représenté par la largeur de la figure (a), Y la hauteur et Z la profondeur du plan (la hauteur de la figure (b)), considérons trois objets s'éclipsant les uns les autres :

— En bleu, un objet simple largement occulté ;
— En rouge, un objet simple principalement occultant ;
— En vert, un objet complexe occultant et occulté.

Nous cherchons à segmenter trois calques, représentants trois aplats de textures, suivant le plan de la figure (a). Une première approche pourrait consister à réaliser une segmentation des plans avec la méthode de RANSAC. Entre chaque point ou pixels, les distances sur les coordonnées et l'intensité peuvent enrichir la segmentation avec un vote pondéré favorisant la distance des points, en mètres, à une variation brut sur le ou les histogrammes des pixels pouvant indiquer une occultation ou un contour. L'aplat des textures par calque correspondrais alors aux figures (c), (d) et (e). Ensuite, nous labeliserions automatiquement et approximativement des régions (en jaune) pour Inpainting. Les limites des labels pourrons êtres estimés selon, par exemple, de la reconnaissance de formes basée sur la connaissance des objets attendus, par une simple reconstruction médiane ou encore en réalisant une propagation répondants aux orientations locales des zones supposées occultées (très fort changement sur histogramme et manque d'information entre deux zones sur un même calque). Les objets occultés seraient alors reconstruis sur leurs calques. Pour finir, nous découperions automatiquement les images de façons à limiter la présence du fond des calques, qui ne contiendrait aucune information. Nous proposerions alors de remplir ce dernier par une texture générée aléatoirement afin d'éviter que l'étape de reconnaissance ne se base dessus. Cette dernière étape mise sur l'utilisation d'un réseau de neurones convolutif (avec autant d'améliorations que souhaité). Pour une acquisition, nous passerions tous les calques extraits au réseau de neurones (trois pour une image dans notre exemple schématique) et ce dernier reconnaîtrai les éléments segmentés avec une bonne précision (d'après quelques tests que nous avions réalisés partiellement visibles dans la partie 2.1.6.4).

La méthode que nous venons de décrire accepterais la présence de parasites en nombre respectable et une reconstruction imprécise. Les cas où la reconstruction trahirait l'aspect de l'objet sont en général les mêmes cas où l'homme ne peut que supposer sa forme. La méthode pourrais donc être améliorée par une approche de segmentation de plus en plus cognitive, permettant de mieux reconstruire les formes basiques des objets (exemple : un ballon occulté à 80% par un mur peut être reconstruit comme sphérique par une approche cognitive là où une approche sur les orientations locales aura tendance à reconstruire une forme au mieux en 'D'). La connaissance de la segmentation serais donc d'aspect général (formes basiques, segmentation d'objets supposés) quand la connaissance de l'étape de reconnaissance serais déterminante et précise.



(a) vue dessus (b) Vue de côté (c) Calque a (d) Calque b

(e) Calque c (f) Calque a Inpaint (g) Calque b Inpaint (h) Calque c Inpaint

FIGURE 18 – Segmentation intelligente

2.1.6.5 SVM : Machine à vecteurs de support

Utiliser des machines à vecteurs de support peut s'avérer efficace pour classifier certains objets [21, 17]. Ils sont notamment très utiles pour établir une reconnaissance intra-classe sur les groupes génériques (reconnaître le type de fleur depuis le fait déterminé que l'objet traité est une fleur). En se basant sur un vecteur de caractéristique(s) composé de descriptions numériques représentant nos objets, nous pouvons avoir une approche différente pour l'établissement de la similarité et par conséquent, enrichir les précédentes. Un SVM ou des méthodes d'apprentissage classiques (voir Annexe [1], État de l'art partie 1.4.2) nous sera plus utile pour segmenter des points d'intérêts locaux ou des groupes d'orientations locales (voir partie 2.1.6.2).

3 Analyse et segmentation de structures dans les images

3.1 Projet véhicule autonome

Au cours de la recherche et du développement de solutions de reconnaissance d'objets, d'autres problématiques de traitement d'images ont vu le jour sur des projets parallèles. Nous traiterons ici de l'intégration d'un service de segmentation supervisé dont les plus-valus sont le gain de temps, la précision de la segmentation et la ré-utilisabilité.

3.1.1 Problématique de segmentation

Nous souhaitons établir l'espace de liberté d'un véhicule autonome depuis des images satellites. Nous devons par conséquent établir un masque binaire séparant les zones accessibles et ouvertes des zones accidentées et impraticables. Divers algorithmes apportés par l'équipe s'occuperont ensuite de gérer les déplacements du système. Pour la tâche de segmentation, nous pouvons faire deux propositions :

A) **Segmentation automatique** : Le cas idéal, la segmentation automatique de l'image indiquerait par une reconnaissance de textures, couleurs et contours les zones accessibles des zones non accessibles. La nature des terrains à traiter (herbe, forêts, maison avec jardin, chemins et routes) n'étant que très peu caractéristique (présence de ces éléments dans les deux espaces), le gradient de l'image étant difficile à interpréter et le besoin de précision étant critique, nous ne pouvons pas proposer une solution suffisamment viable pour la segmentation automatique.

B) **Segmentation supervisée** : Le problème de précision et d'interprétation sémantique des lieux peut être résolu par un humain, notamment car cette opération n'a lieue qu'à l'initialisation du système et est pseudo-définitive (la segmentation n'a de raison de changer qu'au changement des lieux mêmes). À l'aide d'outils, on supervise un algorithme de segmentation. L'orientation algorithmique semblant la plus pertinente est alors une méthode de **propagation de labels**. Le superviseur labelise grossièrement les zones de libertés des zones impraticables et une propagation sensible aux bords de ces zones et aux autres labels a lieu. Après sélection du label correspondant à l'espace de liberté, nous pouvons enfin éroder notre masque binaire pour établir une marge de confort aux bordures des zones impraticables et praticables.

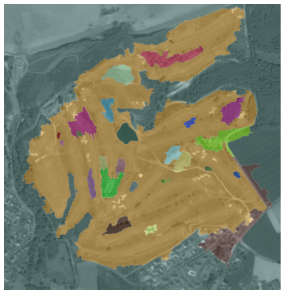
La seconde solution étant la plus appropriée à la problématique (une solution simple et rapide pour des résultats certains), elle fut implémentée. Il existe plusieurs méthodes de propagation de labels [24, 13, 25], nous utiliserons une méthode basée sur l'idée du Watershed[5].

3.1.2 Segmentation supervisée

(a) Image d'origine

(b) Supervision des labels

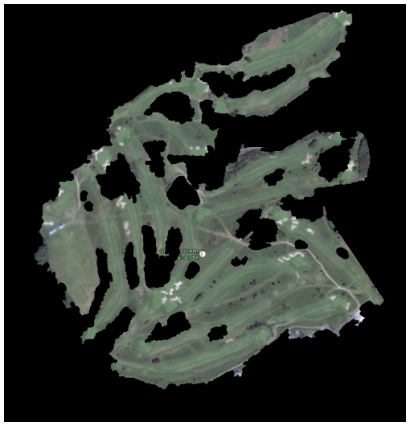
(c) Watershed



(d) Source/Segmentation

(e) Masque binaire obtenu

(f) Érosion du masque



(g) Extraction pour démonstration

FIGURE 19 – Segmentation supervisée de zones praticables

D'autres exemples et démonstration sont disponibles en annexe 5 (figures 54 et 53).

3.1.3 Voies d'amélioration

Idéalement, nous recherchions une méthode de segmentation automatique de terrains à couleurs et textures similaires. Une intuition m'indique qu'il est possible d'**automatiser**

la labellisation de ceux-ci. Cela serait rendu possible en utilisant un réseau de neurones orientés sur cette problématique. On aurait pu alors penser à *TerraPattern* (`http://www.terrapattern.com/` qui, avec un réseau de neurones résiduel basé sur ResNet[12], a appris entre autre à différencier les différents types d'espaces verts. En réalisant une extraction locale glissante de l'image par blocs, nous demanderions à un réseau similaire de labeliser chacun des blocs extrais suivant sa meilleure reconnaissance (un type de label par classe). La structure des labels sur le masque serait alors très différente de la méthode supervisée, nous aurions un point par bloc et la propagation se fera depuis une grille de point. Ainsi la précision des labels et des cas particuliers sera inversement proportionnelle à la taille du bloc ; il n'y aura pas d'oublis mais aucun effort d'adaptation pour la segmentation sémantique des cas particuliers ne sera effectué. Enfin, les résultats d'une telle approche reste à démontrer et la solution est coûteuse à mettre en place pour une tâche d'initialisation qui n'est nécessaire qu'une seule et unique fois ; ainsi, bien qu'elle soit intéressante, nous n'implémenterons ou ne testerons pas cette méthode.

D'autres part, nous pouvons aussi améliorer la segmentation en optant pour une autre approche de propagation de label ou en implémentant une amélioration des solutions courantes afin de limiter les interactions nécessaires par l'utilisateur [8] ;

3.2 Extraction de structure et contenu dans des documents

Plusieurs tâches d'implémentation de solutions pour l'extraction non supervisée de données dans des documents sont attribuées au stagiaire ; ces besoins sont à l'origine du développement de la bibliothèque Structure Extraction for Recognition of Visual Elements in Documents (SERVED).

3.2.1 Suggestion de titre

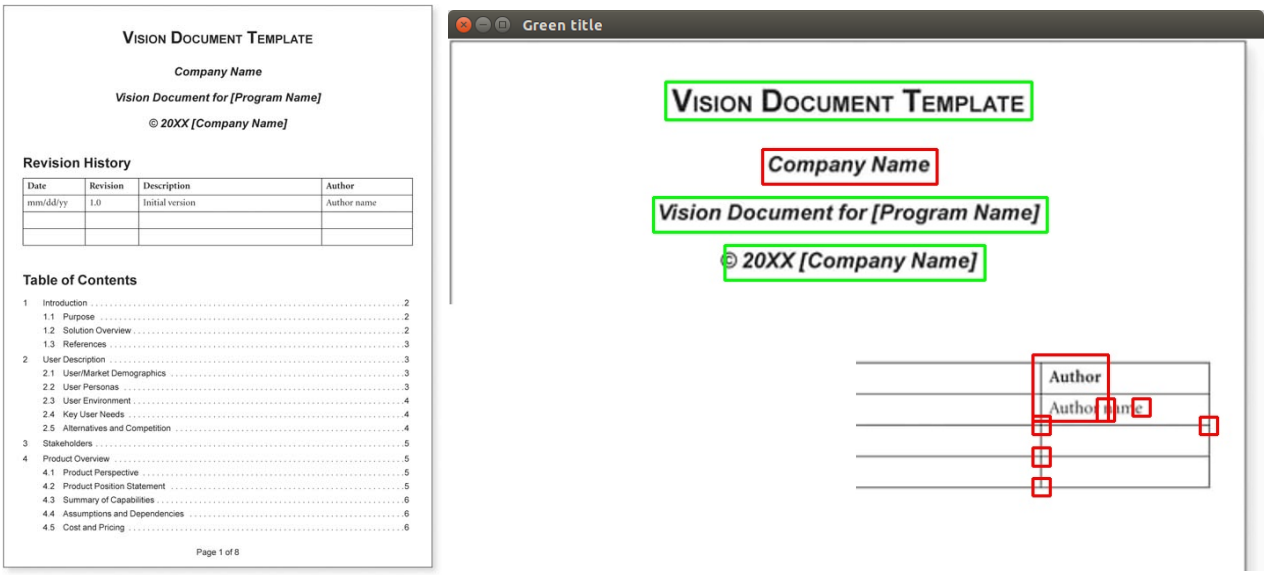
3.2.1.1 Problématique et solution

Cordon D.S.2I possède une plateforme permettant de visualiser un grand nombre de documents professionnels. Ces documents sont stockés sous forme d'images après avoir étés scannés et ne possèdent pas tous de nom ; on cherche donc à les titrer. Pour ce faire, nous souhaitons établir un système d'extraction et reconnaissance visuelle de titre. L'état d'esprit pour la détection est le suivant : un résultat parasité ou moyennement fidèle est davantage acceptable que les titres actuels qui sont vides ou aléatoires.

La nature des documents présents dans la base d'images simplifie la tâche d'extraction : les titres sont positionnés dans la partie haute des documents et obéissent généralement à une hiérarchie ; ils sont donc souvent plus gros et plus centrés que le reste des informations. Nous nous basons sur ces propriétés pour définir des régions d'intérêts pour la présence de titre. En commençant par débruiter ces parties du document par moyennes non-locales [2], nous en segmentons les éléments à l'aide d'une binarisation adaptant son seuil par régions afin de pouvoir traiter les zones potentiellement ombrées (relation entre l'éclairage du scanner et les coins du document ou les reliures). Nous utilisons alors des opérateurs morphologiques simples sur notre première extraction pour faire disparaître les détails fins (bruit persistant,

agrafes, etc.) à l'aide d'une ouverture "violente" (c.a.d avec une dilatation environ quatre fois plus grande que l'érosion) couvrant ainsi les différents éléments du document, fusionnant les sous-ensembles proches (caractères d'un mot) afin de finalement construire le masque d'extraction des informations. Nous conservons alors naïvement les plus grandes zones d'intérêts en tant que candidats à la constitution du titre. Celles-ci (et celles-ci uniquement pour des soucis d'optimisation) sont traitées avec l'OCR Tesseract (voir partie 4.1.9). SERVED est alors capable de retourner le titre formaté selon des règles simples (au choix et combinable : élimination de caractères aberrants, de répétitions de caractères spéciaux, de retour à la ligne, etc.) et un seuil de confiance définissable par l'utilisateur (par défaut 50%). La méthode donne des titres de document convaincants, elle est cependant limitée à :

- L'écriture numérique sur des documents professionnels uniquement,
- La nécessité d'avoir des documents de résolution acceptable (la précision de la reconnaissance avec l'OCR est directement liée à la lisibilité des caractères) avec une police compacte (un italique très fin et très espacé comme dans la figure 57 des annexes risque d'être détruit par l'ouverture morphologique ou d'être segmenté),
- quelques secondes (3 pour un document moyen sur le serveur utilisé) par extraction (ouverture, pré-traitement, sélection et reconnaissance),
- Une sélection pertinente mais naïve des régions d'intérêts qui ne permet pas de reconnaître le titre de document à la couverture unique ou exotique.



(a) Document original

(b) Pré-traitement

FIGURE 20 – Suggestion de titre :
E ZOXX [Company Name]-VISION DOCUMENT TEMPLATE-Vision Document IorIProgram Name]

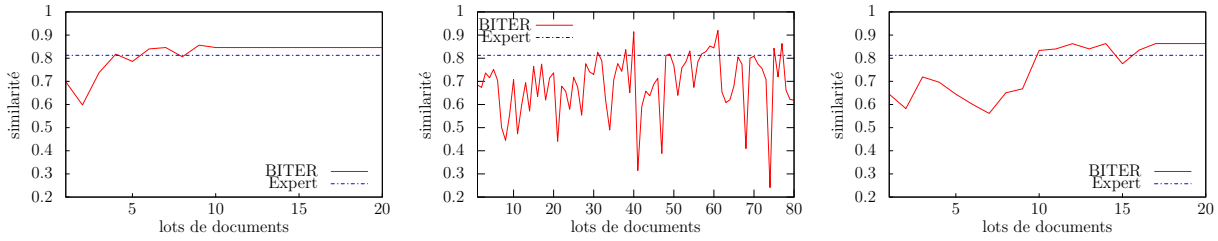
3.2.1.2 Apprentissage par renforcement

Les paramètres des opérations de pré-traitement à la reconnaissance optique de texte ont été établis par expertise et par observations issus de tests. Ceux-ci, qui sont souvent des

coefficients, des amplitudes, des tailles de patchs ou des types d'opérations, ne permettent pas l'extraction de titre sur des cas particuliers (couverture illustrée, titre en bas de page, …). Ainsi et pour généraliser la méthode au plus grand ensemble de documents, on accepte, dans une certaine mesure, du bruit et des parasites. On s'intéresse donc au renforcement du choix des paramètres de SERVED par type de documents.

L'idée d'améliorer la reconnaissance de titre depuis un document grâce à l'apprentissage provient de mon tuteur professionnel de stage. Nous avons mis à profit le travail [18] d'E.NICART, doctorante au moment de la rédaction de ce rapport et auteur de la plateforme d'intelligence artificielle : BIMBO (***B****enefiting from* ***I****ntelligent and* ***M****easurable* ***B****ehaviour* ***O****ptimisation*) afin de renforcer le paramétrage des critères de segmentation et d'extraction des zones d'intérêts pour la suggestion de titres. L'objectif est alors l'amélioration continue de ces suggestions en fonction du feedback des utilisateurs et du type de document. Ce dernier est déterminé par sa taille (indiquant une nature portrait ou paysage) ainsi qu'un indicateur faible de complexité (basé sur la moyenne de l'histogramme issu de l'image ramenée en niveau de gris) indiquant une nature professionnelle ou illustrée de la page de titre. Ainsi et avec l'aide de G.LEROY, lui aussi stagiaire dans l'entreprise, BITER a été intégré.

BITER (***B****imbo's* ***I****ntelligent* ***T****itle* ***E****xtraction and* ***R****ecognition*) propose des profils de segmentations depuis des listes de paramètres cohérents. Elle fait varier autant de paramètres qu'elle souhaite et étudie les conséquences de ces changements sur le résultat de l'extraction en le comparant à la vérité terrain. On utilise une moyenne pondérée d'un *Fuzzy Levenshtein* avec la confiance de l'OCR en l'extraction (cette seconde partie permettant de discriminer les positifs issus du hasard) pour le calcul de la distance de similarité. Nous avons construit un corpus de 42 documents dont le titre attendu a été identifié par l'homme. L'IA a été récompensée ou pénalisée sur la similarité de ses extractions avec les titres attendus. Les premiers résultats sont encourageants :



(a) Epsilon=0.2/2 tous les 84 documents (2 lots)

(b) Epsilon=0.2/1.05 tous les 84 documents (2 lots)

(c) Epsilon=0.2/1.5 tous les 84 documents (2 lots)

FIGURE 21 – Premiers résultats pour la suggestion de titre renforcée avec BITER

Sur les graphiques ci-dessus, epsilon définit la curiosité de l'algorithme, sa susceptibilité à explorer des combinaisons de paramètres. Plus epsilon est petit, plus elle utilise le profil qui a déjà donné la meilleure récompense. On remarque sur 21a que l'apprentissage par renforcement surpasse l'approche experte mais aussi qu'il se stabilise trop rapidement à 84,6%. Cela est issu de la diminution d'epsilon trop importante dans le temps. Pour lui laisser l'opportunité d'améliorer ces résultats, nous réduisons le facteur de diminution, voir figure 21b, et on observe que l'algorithme explore trop et ne converge pas rapidement. Finalement, avec

diminution de 150% tous les deux lots de documents, figure 21c, on converge vers des profils de segmentation et d'extraction avec une meilleure précision que celle de l'expert : 86.3% contre 81.2%.

3.2.2 Extraction et reconnaissance d'éléments visuels structurants

On cherche à présent extraire les éléments d'un document, les localiser et les identifier. Il est requis, depuis une image de celui-ci, d'extraire la position et la description d'un ensemble logique (rectangle englobant décrivant un paragraphe, graphique, QR code, code barre, signature etc.).

3.2.3 Structuration du document

On choisit de structurer le document suivant un schéma simple : un document est composé d'éléments d'un certain type, associés à une valeur et une localisation. Concrètement SERVED permet trois étapes :

A) **Segmentation visuelle des éléments**, à l'aide de méthodes d'amélioration de l'image ainsi que de détection de contours appliquée à des opérateurs morphologiques simples similaires à ceux utilisés dans la partie 3.2.1. On initialise une description en JSON de l'image passée en paramètre en renseignant les rectangles englobant des éléments découverts dans le format suivant :

```
{
  "imageSample9" : [
    {
      "type" : "undefined",
      "value" : "Undefined",
      "bounding-box" : [
        [
          28,
          662
        ],
        [
          385,
          793
        ]
      ]
    },
    etc...
}
```

B) Non implémenté : **Reconnaissance générique des types d'éléments**, de façon visuelle, à l'aide d'un réseau de neurones simple, afin de renseigner le type de l'élément dans notre description.

C) Non implémenté : **Reconnaissance du contenu des éléments à l'aide de systèmes dédiés**.

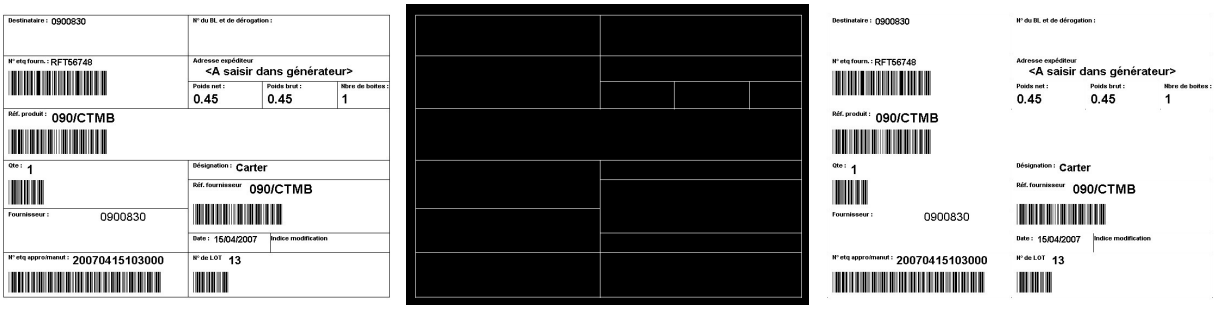
(a) Document original (b) Segmentation (c) Structure Displayer

FIGURE 22 – Extraction et initialisation des éléments d'un Document (description JSON en annexe 2)

3.2.4 Segmentation de tableaux et séparateurs

Les tableaux font partis des éléments les plus simples à segmenter, il s'agit de faire de la reconnaissance d'axes horizontaux et verticaux. Même si la transformée de Hough semble toute indiquée pour cette tâche, nous ne l'utiliserons pas ; une intuition nous fait penser que la segmentation d'un tableau dépendrait des éléments voisin. Nous utilisons donc des opérateurs morphologiques ainsi qu'une méthode [22] d'analyse topologique pour extraire les contours sur le masque binaire obtenu.

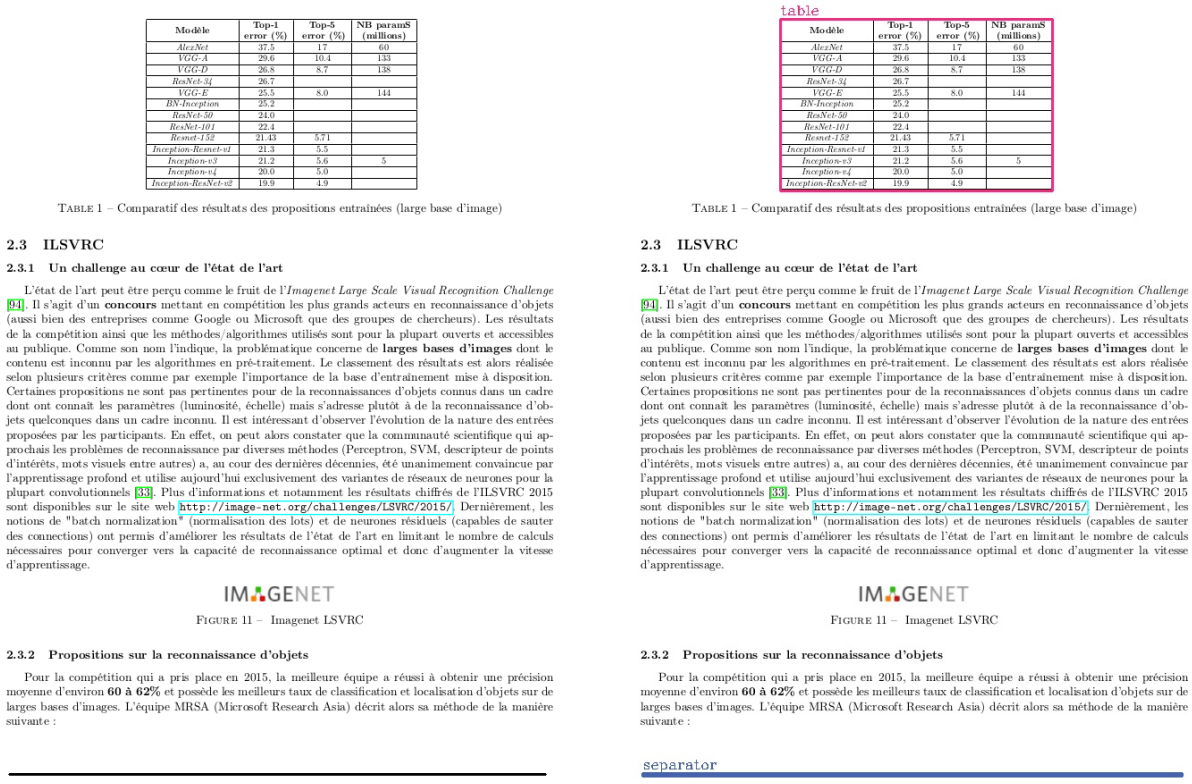
Dans un premier temps, nous réalisons deux ouvertures aux éléments structurants horizontaux et verticaux. Ceux-ci nous permettrons de construire deux masques (respectivement composés des traits horizontaux et verticaux) que nous combinons avec un "OU" logique. La somme des masques extrait permet donc de nouvelles opérations de débruitage (par soustraction) d'éléments.



(a) Tableau original (b) Masque obtenu (c) Tableau nettoyé

FIGURE 23 – Exemple de détection de tableaux pour le débruitage.

La soustraction est efficace uniquement pour les cas de lignes parfaites. Pour les cadres "adoucis ou "anti-aliasés", on effectue un filtrage sur l'image original depuis le masque dilaté pour reconstruire une image sans tableaux. On évite ainsi la soustraction de pixel semi-transparent au coût d'une segmentation plus brute et plus fidèle, mais moins précise. On s'intéresse, dans un premier temps, à retrouver ces objets dans un document. On établit donc les contours sur notre masque :



Modèle	Top-1 error (%)	Top-5 error (%)	NB paramS (millions)
AlexNet	37.5	17	60
VGG-A	29.6	10.4	133
VGG-D	26.8	8.7	138
ResNet-34	26.7		
VGG-E	25.5	8.0	144
BN-Inception	25.2		
ResNet-50	24.0		
ResNet-101	22.4		
Resnet-152	21.43	5.71	
Inception-Resnet-v1	21.3	5.5	
Inception-v3	21.2	5.6	5
Inception-v4	20.0	5.0	
Inception-ResNet-v2	19.9	4.9	

TABLE 1 – Comparatif des résultats des propositions entraînées (large base d'image)

2.3 ILSVRC

2.3.1 Un challenge au cœur de l'état de l'art

L'état de l'art peut être perçu comme le fruit de l'*Imagenet Large Scale Visual Recognition Challenge* [94]. Il s'agit d'un **concours** mettant en compétition les plus grands acteurs en reconnaissance d'objets (aussi bien des entreprises comme Google ou Microsoft que des groupes de chercheurs). Les résultats de la compétition ainsi que les méthodes/algorithmes utilisés sont pour la plupart ouverts et accessibles au publique. Comme son nom l'indique, la problématique concerne de **larges bases d'images** dont le contenu est inconnu par les algorithmes en pré-traitement. Le classement des résultats est alors réalisée selon plusieurs critères comme par exemple l'importance de la base d'entraînement mise à disposition. Certaines propositions ne sont pas pertinentes pour de la reconnaissances d'objets connus dans un cadre dont ont connaît les paramètres (luminosité, échelle) mais s'adresse plutôt à de la reconnaissance d'objets quelconques dans un cadre inconnu. Il est intéressant d'observer l'évolution de la nature des entrées proposées par les participants. En effet, on peut alors constater que la communauté scientifique qui approchais les problèmes de reconnaissance par diverses méthodes (Perceptron, SVM, descripteur de points d'intérêts, mots visuels entre autres) a, au cour des dernières décennies, été unanimement convaincue par l'apprentissage profond et utilise aujourd'hui exclusivement des variantes de réseaux de neurones pour la plupart convolutionnels [33]. Plus d'informations et notamment les résultats chiffrés de l'ILSVRC 2015 sont disponibles sur le site web http://image-net.org/challenges/LSVRC/2015/. Dernièrement, les notions de "batch normalization" (normalisation des lots) et de neurones résiduels (capables de sauter des connections) ont permis d'améliorer les résultats de l'état de l'art en limitant le nombre de calculs nécessaires pour converger vers la capacité de reconnaissance optimal et donc d'augmenter la vitesse d'apprentissage.

IMAGENET

FIGURE 11 – Imagenet LSVRC

2.3.2 Propositions sur la reconnaissance d'objets

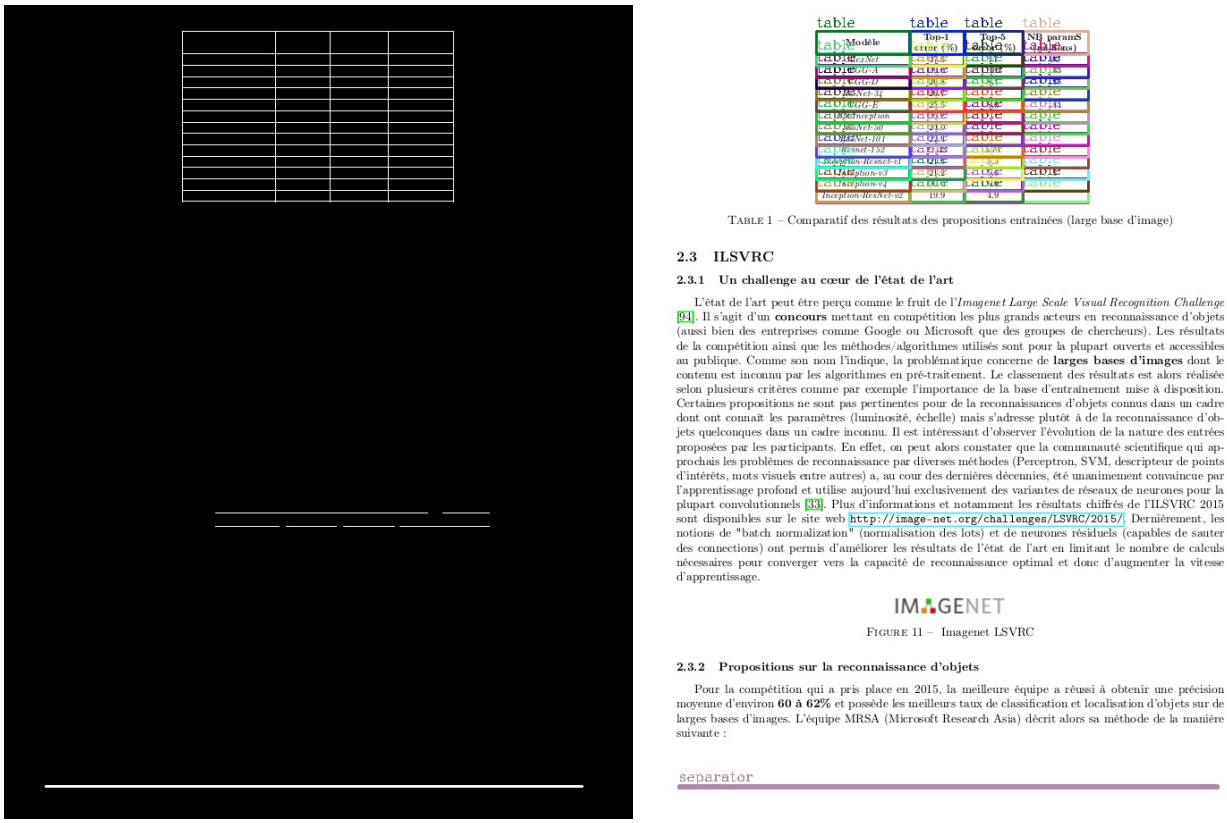
Pour la compétition qui a pris place en 2015, la meilleure équipe a réussi à obtenir une précision moyenne d'environ **60 à 62%** et possède les meilleurs taux de classification et localisation d'objets sur de larges bases d'images. L'équipe MRSA (Microsoft Research Asia) décrit alors sa méthode de la manière suivante :

(a) Exemple de document [15]

(b) Classification de tableaux

FIGURE 24 – Reconnaissance de tableaux dans un document

Sur le document ci-dessus, on décompte deux tableaux, le séparateur de bas de page constituant un faux positif de la segmentation de tableaux. La détection n'a pas lieu sur les cadres à aires restreintes (en vert) et peine sur les cadres trop proches du texte (en cyan). On peut utiliser un "ET" logique afin de labelliser les tableaux des séparateurs ou autre traits isolés. Ainsi, sur nos résultats, tout coin de tableau, représentant l'intersection des lignes horizontales et verticales mis en avant par ce "ET", valide la classification. Cependant, le coût de l'opération étant beaucoup plus élevé qu'une vérification basée hauteur et longueur du rectangle englobant, on préfère cette dernière solution. Dans cette configuration, on filtre les contours détectés selon une hiérarchie d'appartenance. Pour isoler le tableau dans sa totalité, on repère le contour parent des autres contour pour ne garder que celui-ci. Cependant, puisque le masque est fidèle au tableau d'origine, on peut prétendre à extraire l'information de chaque case pour la faire passer à l'OCR.



(a) Les contours sont fidèles

(b) Extraction des cases

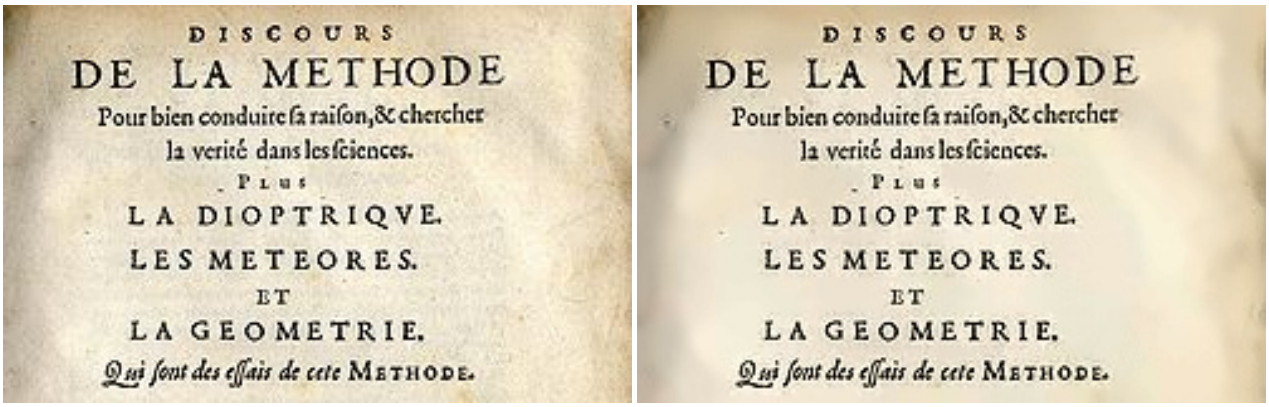
FIGURE 25 – Reconnaissance des cellules d'un tableau dans un document

On peut ainsi utiliser les outils que nous avons développés pour créer une hiérarchie de cellules contenant leurs valeurs textuelles.

3.2.5 Réparation et ajustement des images

3.2.5.1 Non-local Means Denoising

Comme nous en avons brièvement parlé dans la partie 3.2.1.1, les documents sont scannés et donc sujet à des bruits d'origines diverses. La méthode de débruitage [2] que nous utilisons en pré-traitement de la plupart de nos opérations de segmentation permet de supprimer celui-ci. Ainsi, nous évitons les parasites très fin qu'ils soient d'origine physique (poussière) ou d'origine logicielle (e.g correction de contraste par le scanner) sans affaiblir les contours des éléments structurants. La précision de la segmentation des éléments (e.g binarisation adaptative) depuis l'image débruitée sera alors accrue par rapport à l'original et les parasites plus gros (e.g agrafes) ignorés lors des opération morphologiques.



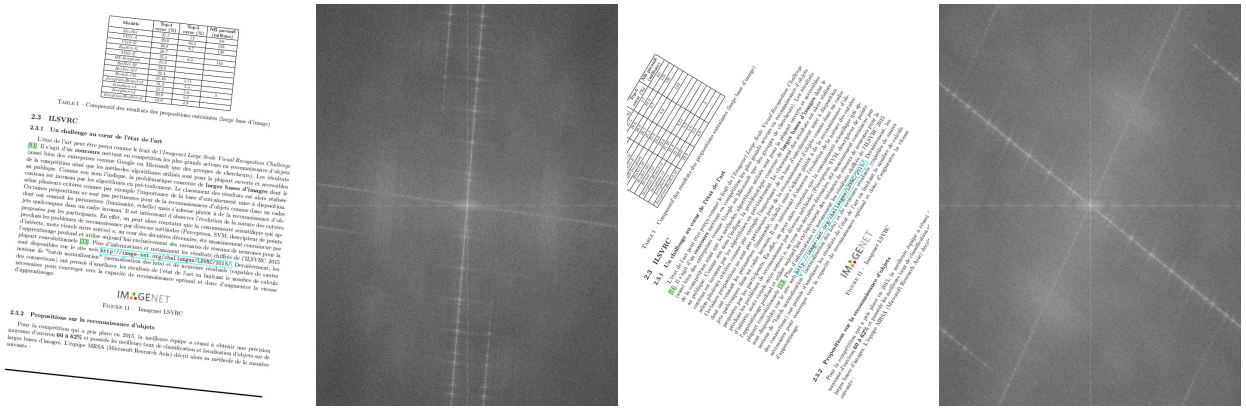
(a) Image originale (b) Non-Local Means Denoising

FIGURE 26 – Réparation d'images parasitées

3.2.5.2 Orientation depuis la TFD

Nos méthodes de segmentation possèdent des propriétés (i.e des noyaux et des critères de sélection) entièrement basées sur l'orientation des éléments d'un document dans l'espace : il est nécessaire que le texte soit aligné avec les bords de l'image. Les corpus que nous étudions est composé d'images ne répondant pas toutes à ce critère ; on cherche à aligner automatiquement ces images.

Nous étudions dans un premier temps l'orientation du document dans sa matrice pour déterminer son orientation générale. Nous nous intéressons donc à la Transformée de Fourier Discrète (TFD) qui, dans notre cas, nous fournit un indicateur fiable de l'orientation :

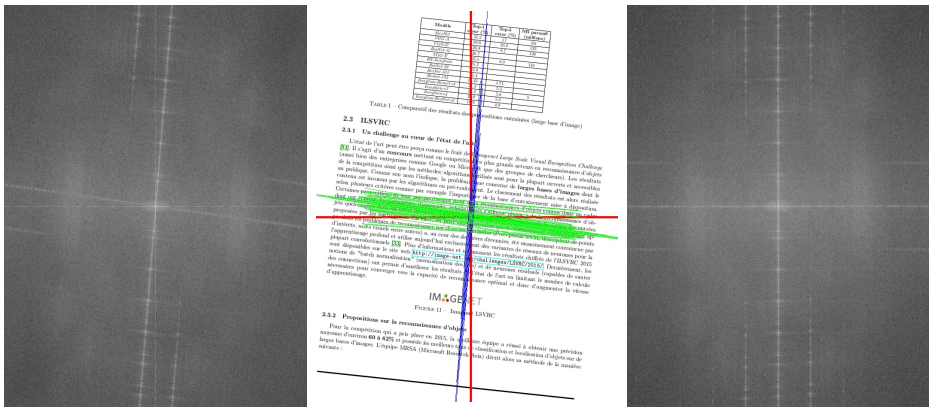


(a) Image A (b) TFD de A (c) Image B (d) TFD de B

FIGURE 27 – Orientation d'une image depuis sa transformée de Fourier discrète

On s'intéresse dès lors aux lignes apparentes de la TFD des images. L'orientation du document est défini par l'angle entre les axes obliques et les axes droits (perpendiculaire aux bords de l'image). On effectue donc un seuillage de la TFD pour conserver les points

aux intensités les plus élevées (les croix) afin d'isoler les lignes grâce à une transformée de Hough. Selon le seuillage et le contenu de l'image, on obtient deux groupes de lignes plus ou moins disparates approchant celles de la TFD. On choisit de faire la moyenne des lignes de chaque groupe puis de calculer l'angle des deux droites moyennes. On réalise alors la moyenne des angles et on applique une rotation à l'image. On notera qu'une structure particulière et prononcée sur l'image ne nous permettra pas d'exploiter cette méthode (voir annexe 59).



(a) TFD d'un document oblique (b) Seuillage et transformée de Hough (c) Realignement depuis les lignes moyennes (d) Extraction réalisée après rotation de 6,27°

FIGURE 28 – Ré-alignement d'une image oblique avec les méthodes de Hough et Fourier

Sur l'exemple ci-dessus, on trouve des angles de -4,96° et -7,48°, on effectue donc une rotation de -6,22° pour ré-aligner le document. Notre méthode de segmentation de tableaux et séparateurs peux ainsi nous retourner des résultats (contre 0 positifs sans la ré-orientation).

Points faibles de la méthode :

— Notre méthode de réalignement ré-aligne notre image sur l'axe le plus proche parmi l'horizontal et le vertical, sur 90°, ce qui signifie :
  — Qu'un document portrait trop mal scanné (angle de 75°, par exemple) serra ré-aligné dans un format paysage,
  — Qu'un document inversé sera ré-aligné dans son cadre, mais non retranscrit dans le sens normal de lecture,
— Elle est entièrement basée sur la mise en valeur des lignes de la TFD, qui peuvent mal apparaître (voir figures 58 et 60 en annexes).

3.2.6 Voies d'amélioration

SERVED est sujet à amélioration, cette partie liste les propositions les plus notifiables.

Les documents peuvent être obliques et nous les réorientons. Toutefois, dans des cas rares, les éléments structurants d'un document peuvent l'être sans que la page ne le soit. Dès lors les coins des rectangles englobants que nous utilisons se superposerons et le même contenu apparaîtra sur plusieurs éléments. Une voie d'amélioration à moindre coût serait d'utiliser des rectangles englobants orientés pouvant ainsi mieux épouser la forme des éléments. Le

contour exact d'un élément pourrait aussi être stocké dans la mesure ou stocker des milliers de points par éléments d'un document serait acceptable et nécessaire. Nous pouvons aussi utiliser le travail réalisé sur les moments des blocs binaires des éléments dans l'image 61 pour mieux les décrire.

4 Informations complémentaires

Cette partie se propose de revenir sur les aspects non détaillés des travaux décrits par ce rapport.

4.1 Techniques

4.1.1 OpenCv, implémentations

La plupart des implémentations algorithmiques ont été réalisées en utilisant OpenCV. Les sujets traités dans les parties 2 et 3.1 utilisent du C++ et la version 2.4 d'OpenCV. Les sujets utilisant SERVED, ont été traités en utilisant Java 1.8 et OpenCV 3.1.0.

FIGURE 29 – Bibliothèque.

FIGURE 30 – Langage.

FIGURE 31 – Langage.

OpenCV pour "**Open** **C**omputer **V**ision" est une bibliothèque graphique libre initialement dévelopée par INTEL et maintenue par la société de robotique WILLOW GARAGE depuis 2008. Aux performances largement optimisées et bénéficiant d'un excellent support de la communauté, cette technologie trouve sa place dans la plupart des solutions de visions par ordinateurs développées à ce jour et s'impose comme un standard dans le domaine de la recherche en traitement d'image.

Les solutions de reconnaissance visuelle dans un flux vidéo ont la particularité de fonctionner sur une boucle infinie mais interruptible de traitements itératifs et interactifs d'images acquises par une caméra en temps réel. Les implémentations sont expérimentales, l'interactivité du programme est donc étendue (touches pour paramétrer et activer certains traitements, curseurs pour les seuiller) et plusieurs affichages comparatifs sont présents. Dans la plupart des cas, l'interface homme-machine est très limitée en raison de la nature des projets portant sur le **traitement automatique** de l'image. Si on souhaite rendre nos systèmes flexibles à plusieurs environnements, nous souhaitons toutefois passer par l'étape de paramétrage une seule et unique fois, à l'initialisation du système dans son environnement. L'interactivité des combinaisons d'algorithmes expérimentaux est expliquée dans l'aide interne des executables

(accessible sur l'environnement lançant l'application, le terminal, en pressant la touche 'h').

Les solutions de segmentations dans des images simples existent en deux versions : une dédiée au développement et une améliorée à des fins de déploiement. La première montre les résultats progressifs des différentes méthodes et algorithmes appliqués aux images (tel qu'on peut les voir dans les images de ce rapport) quand la seconde se contente de fournir un résultat dans un temps plus optimisé.

Enfin, en parallèle et pour automatiser certains aspects des solutions développées, d'innombrables scripts en langages de commandes (de conformation et préparation de base d'images, de déploiement, d'évaluation des résultats etc.), de tests expérimentaux utilisant Pandore ainsi que de fichiers de configuration (d'un RDN en Protobuf, par exemple) ont étés conçus. Nous n'en parlerons pas plus en raison de leurs trop grande spécificité.

4.1.2 Robot Operating System

Nous avons présenté des sous-systèmes de visions dans les différents projets précédemment décrits. Ceux-ci se situent au cœur des projets traités, mais sont destinés à être intégré dans un système plus grand. Ainsi, nous profiterons de cette partie pour parler brièvement de Robot Operating System (ROS).

ROS est un framework largement utilisé en robotique et proposant des bibliothèques et des outils puissants. Le concept derrière ce système est de rendre possible l'échange de plusieurs programmes informatiques tout en conservant leurs indépendances. Ainsi, un ROS est composé de plusieurs sous-systèmes qui pourront facilement être adaptés à l'environnement qui les accueillent. Un des avantages de ROS est que rien n'empêche de partager les sous-systèmes avec d'autres projets, permettant ainsi de reproduire un savoir faire avec un minimum d'efforts. A titre d'exemple, dans le cadre d'un système de vision par ordinateur, l'ensemble de reconnaissance que nous implémentons peut être réalisé dans un langage de programmation différent du reste du système (qui comportera, par exemple, une couche de communication avec un serveur) et utilisé dans des projets futurs. De ce fait, la portée du travail réalisé et des résultats obtenus au cours du stage se retrouvent naturellement accrues.

Figure 32 – Robot Operating System

4.1.3 TensorFlow

TensorFlow est un outil open-source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache. Nous l'utiliserons pour effectuer des approches de reconnaissance visuelle d'objets avec des réseau de neurones, particulièrement avec Inception-v3.

FIGURE 33 – Bibliothèque d'apprentissage automatique

4.1.4 PCL

***P**oint **C**loud **L**ibrary* est une bibliothèque ouverte de traitement de nuages de points. Elle permet, entre autre, de travailler avec des scènes incluant la profondeur pour la reconnaissance d'objets en trois dimensions.

FIGURE 34 – Bibliothèque de traitement de nuages de points

4.1.5 OpenNi

***Open** **N**atural **I**nterraction* est une organisation et une solution Open-Source pour faciliter l'utilisation des appareils d'acquisition de profondeurs tels que Structure Sensor, Kinect ou encore Xtion.

FIGURE 35 – Solution logiciel pour l'acquisition de scène 3D

4.1.6 Kinect

Kinect pour "*kinetic*" et "connect" (respectivement de l'anglais "cinétique" et "connecter") est un périphérique de Microsoft permettant l'acquisition de scène en Red Green Blue Depth (RGBD). Elle possède des capteurs de couleur, de profondeur ainsi qu'un micro et un moteur pour ajuster l'angle d'acquisition. Nous avons utilisé une kinect première génération pour certains de nos travaux, elle possède un champ de vision horizontal de 57 degrés (représenté sur 640 pixels) pour un champ de vision vertical de 43 degrés (480 pixels). Ses capteurs ne dépassent pas les 30 images par secondes et la détection est optimal entre 1,2m et 3,5m de distance.

FIGURE 36 – Solution matérielle pour l'acquisition de scène 3D

4.1.7 ConvNetJs

La disponibilité de plusieurs postes informatiques dédiés au stage a permis de réaliser des expériences par curiosité. Celles-ci, pour la plupart non nécessaires mais intéressantes et formatrices, sollicitent les ressources d'une des machines et n'entravent en rien le travail effectué en parallèle. Ainsi, nous avons pu entraîner un réseau de neurone convolutif simple (3 convolutions toutes accompagnées d'une rectification linéaire et d'un vote) sur l'espace d'un mois et observer son évolution ainsi que sa connaissance grâce à *ConvNetJs*. Il s'agit d'une interface Javascript dédiée à l'entraînement d'algorithmes d'apprentissage profond. Dans notre cas, elle nous permet d'observer les entrées et les sorties des couches de neurones mais aussi de constater à tout instant les validations des résultats ainsi que la progression dans le temps de la précision de l'ensemble et des poids attribués. De ces faits, l'utilisation de cette technologie nous a permis de mieux appréhender les apports de chaque élément des réseaux de neurones et d'avoir une idée visuelle de ce que l'ordinateur peut percevoir. L'entraînement a été réalisé sur la base d'image CIFAR-10, constituée d'image à trois canaux (RGB) de 32 pixels de largeur/hauteur représentant des objets ou des animaux non occultés occupant une grande partie de l'image.

FIGURE 37 – Outil de Deep Learning en Javascript

4.1.8 Caffe

Convolutional Architecture for Fast Feature Embedding TODO

Caffe est un framework de deep learning conçu pour être rapide, compréhensible et modulaire. Il est développé par le Berkeley Vision and Learning Center (BVLC) et par des contributeurs communautaires. Yangqing Jia est à l'origine de ce projet qu'il a créé pendant sa thèse à l'université de Berkeley. Au cours du stage, nous nous sommes aidés de Caffe et de son Model Zoo (base de modèles de réseaux de neurones) pour réaliser des tests et des comparaisons sur les réseaux de neurones convolutifs. Il constitue, par ailleurs, un excellent support pour la conception et la compréhension des réseaux de neurones. Davantage d'informations sont disponibles sur `http://caffe.berkeleyvision.org/`.

4.1.9 Tesseract

Tesseract est un moteur de reconnaissance optique de caractères (OCR) originellement développé dans les laboratoires Hewlett-Packard entre 1985 et 1994. Il est ensuite rendu Open-Source en 2005 et est maintenue depuis 2006 par Google. Ce moteur nous permet notamment d'extraire l'information textuelle d'une image en lui associant une confiance de détection. Lorsqu'utilisé sur des images pré-traitée avec OpenCv, les erreurs de détection sont faibles et les résultats suffisamment fiables pour être exploités sur des problèmes de reconnaissance automatisés (e.g extraction de titre d'un document). Nous l'utilisons avec le wrapper Tess4J pour une implémentation utilisant Java.

4.2 Méthodes

4.2.1 Réseaux de neurones

Appliquer un réseau de neurones à un cas de reconnaissance visuelle implique de laisser le choix des caractéristiques discriminantes à l'algorithme. Ce dernier va apprendre à reconnaître un objet depuis la banque d'image sur laquelle il s'est entraîné. Ainsi, un mauvais apprentissage mène à des résultats indésirables.

4.2.1.1 Construction d'une base d'entraînement

L'élaboration d'une base d'entraînement a été prise en charge par le stagiaire à des fins expérimentales sur la reconnaissance des câbles. La nature et la composition de la base sont d'influence critique sur les résultats. Ainsi, il convient de bien construire la banque d'image et de bien appréhender la réaction du réseau afin d'éviter un cas similaire à celui de l'histoire supposée des Tanks du Pentagon [11].

- **Taille** : Plus une matrice est grande, plus son évaluation demande de ressources. Les caractéristiques dominantes d'un objet peuvent être extraites depuis de petites images, de plus, travailler sur une image réduite permet éventuellement de passer outre les artefacts d'acquisition. On choisira de travailler sur des matrices dont la taille ne dépassent pas les 180000 pixels pour un cadre préférentiel de 300x300 pixels.
- **Nombre** : Afin d'éviter d'avoir une attache aux cas particuliers trop importante, le réseau doit être entraîné sur une large base d'images différentes. Une centaine d'images semble alors être un minimum.
- **Fond et transparence** : Les images ne devront pas utiliser de canal alpha. En effet, si l'homme est capable d'interpréter la transparence et de segmenter mentalement l'objet du fond transparent, le réseau de neurone interprétera cette transparence comme une caractéristique descriptive. Par ailleurs, ce fond risque d'être traité comme une couleur unique (du noir par exemple) et gêner la reconnaissance. En effet, il faut éviter, tant que possible que le fond soit à l'origine de la création d'une caractéristique déterminante (fond blanc, par exemple).
- **Aspects colorimétriques** : Nous fournissons des images couleurs en RGB. Le but étant d'entraîner le réseau sur un jeu de données que l'homme aurait pu utiliser lui-même. On souhaite donc laisser la couleur influencer sur le paramétrage du réseau quand bien même celui-ci convertirais les images en noir et blanc.
- **Parasites** : La présence d'objets parasites (sachet plastique, table, cordon d'attachement) n'est pas un problème, ils peuvent même robustifier la connaissance dans la mesure où ils ne sont pas récurrents.
- **Représentation** : Le cadre, l'angle et l'éclairage de l'acquisition influerons sur l'apprentissage. Des milliers d'images d'un objet vu de dessus ne permettent que très rarement la constitution de la connaissance nécessaire à la reconnaissance de ce même objet vu de côté. Il faudra donc au choix fournir des images de l'objet sous tous ses angles ou uniquement dans un angle proche de celui utilisé pour la reconnaissance.
- **Compression** : Certains formats de compression fonctionnent par blocs ; ceux-ci sont souvent invisibles à l'œil nu mais facilement mis en évidence par certains gradients. Il faudra donc faire très attention à ne pas intégrer, dans la base d'image, des motifs de

blocs récurrents susceptibles d'être appris par le réseau.

D'autres détails, notamment sur l'apprentissage d'objets inconnus, peuvent être perçus sur la documentation de TensorFlow [1].

4.2.1.2 Inception, modèle et évolution

Comme nous l'avons vu précédemment (partie 2.1.6.4), Inception-v3 est une architecture de réseaux de neurones proposée par GOOGLE dont l'architecture fut récemment repensée [23] afin de limiter le nombre de paramètres et la complexité des convolutions nécessaires tout en évitant le calcul de bottleneck insignifiants. Nous avons effectué du transfer learning depuis une version pré-entrainée de ce modèle en utilisant TensorFlow pour ajuster la dernière couche de ce réseau et l'appliquer à la détection d'objets définis (télécommande, câble RJ11, chargeur et adaptateur téléphonique).

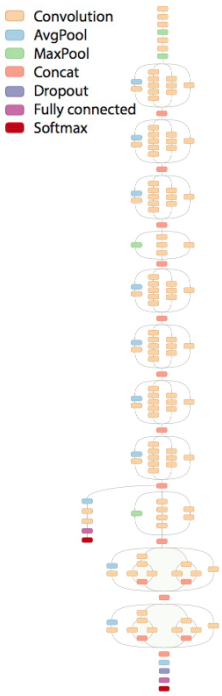


FIGURE 38 – Inception-v3

Pendant le stage, ce modèle a évolué plusieurs fois en intégrant notamment la notion de réseau résiduel. Le modèle d'Inception-ResNet-v2 comprend désormais des raccourcis, lui permettant ainsi d'offrir de meilleurs performances par le biais de réseaux plus profond. Il atteins ainsi une précision de 80,4% (+2,4%) sur son Top-1 résultat et une précision de 95,3%(+1,4%) sur son top-5.

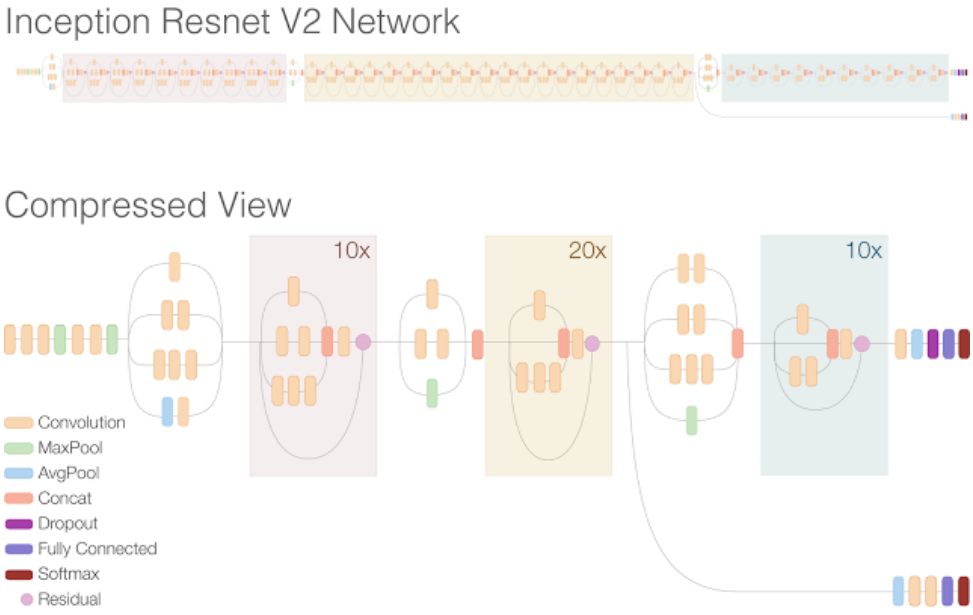


FIGURE 39 – Inception-ResNet-v2

4.2.2 Caractéristiques des images et environnements d'acquisition

Les environnements d'acquisition utilisent soit une Kinect (voir partie 4.1.6) soit une webcam aux capacités de capture d'images similaire. Nous évitons de travailler avec des capteurs trop haute résolution car ils n'enrichissent pas nécessairement la reconnaissance et requièrent beaucoup plus de ressources pour être traitées (puissance et temps de calcul). À titre d'exemple, la base d'image CIFAR-10 (accessible gratuitement sur `https://www.cs.toronto.edu/~kriz/cifar.html`), très utilisée pour tester les approches de reconnaissance, ne propose que des images de 32x32 pixels. Ainsi, l'essentiel des informations nécessaires à la reconnaissance ne demeure pas dans le détail ; nous pouvons donc nous en passer.

Nos environnements d'acquisitions disposent d'un capteur en surplomb des scènes à reconnaître. Le champ d'acquisition, orienté vers le bas, acquiert des objets au repos. L'angle d'acquisition, par rapport au fond de la scène acquise, peut être variable sur les acquisitions 2D mais on préférera positionner nos capteurs de profondeur à la perpendiculaire de celui-ci (puisque nous n'en utilisons qu'un seul et que cet angle permet généralement moins l'occultation grâce à la gravité).

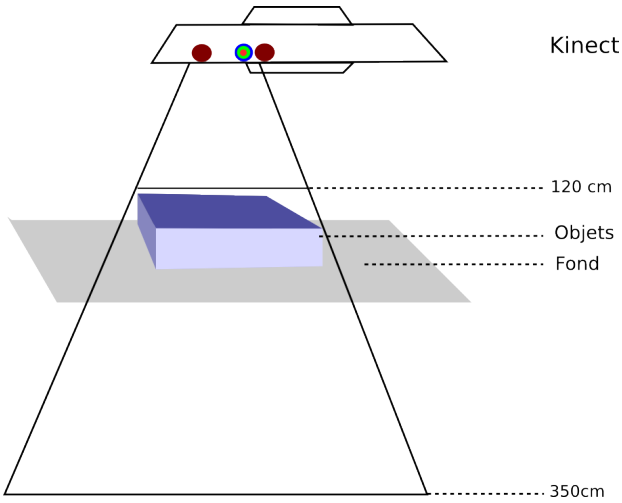


FIGURE 40 – Schéma de l'environnement pour l'acquisition de profondeur

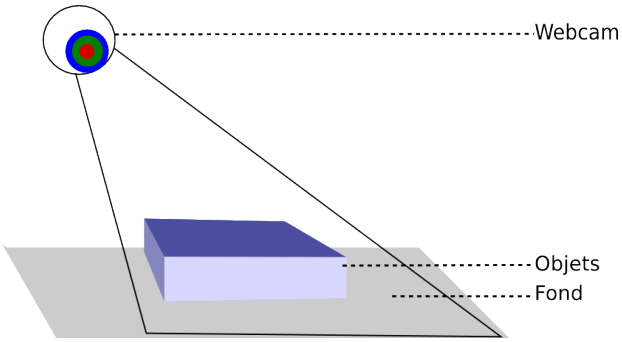


FIGURE 41 – Schéma de l'environnement pour l'acquisition normal

5 Bilan

5.1 Évolution du stage

5.1.1 Évolutions parallèles

Un point intéressant et opportun des objectifs du stage est qu'ils demandaient la rédaction d'un état de l'art tandis qu'en parallèle, certaines conventions, compétitions et forum autour de la vision par ordinateur prenaient place dans le monde. J'ai donc été amené à suivre l'évolution de ceux-ci, tant autour des papiers publiés pour l'occasion que des avis et des résultats qui en émanaient. De fait, même sans cibler ces conventions en particulier, la communauté scientifique avance constamment : j'ai pu témoigner d'un cas où une publication intéressante et à l'état de l'art autour des réseaux de neurones que j'avais lu au début du stage s'est faite surpassée deux fois dans l'espace de six mois. Il est très impressionnant, mais surtout formateur, de voir ces propositions de méthodes et architectures s'améliorer au fil du temps et d'en comprendre les évolutions. Pour finir, j'ai moi-même eu la chance d'assister à un événement rassemblant des présentations de travaux autour du "Deep Learning" dans les images grâce à la journée organisée autour du sujet par l'axe DAC de NormaStic le 28 juin 2016. Cette opportunité m'a permis de mieux comprendre les subtilités du domaine et d'observer les interactions entre les chercheurs participants.

5.1.2 Difficultés rencontrées

Les objectifs du stage m'ont amenés à réaliser de la recherche et développement. J'ai donc été confronté à des problématiques plus ou moins commodes : Certaines relevaient directement de mises en application de connaissances acquises par mes études quand d'autres s'adressaient à ma capacité à élaborer de nouveaux systèmes de vision (une méthodologie nouvelle depuis des sous-ensembles méthodes connues). De ce fait, lors des phases d'expérimentations et d'implémentations, j'ai souvent eu l'occasion de me retrouver dans des semblants d'impasses, face des déceptions comme des surprises en réponse à mes idées. En effet, j'ai pu travailler sur différentes problématiques et ma réponse pour certaines était, malgré les efforts, qu'il n'existe pas de solution viable (ou en tout cas, pas dans l'espace de liberté de celle-ci et que par conséquent, il fallait se résoudre à changer un critère si on souhaitait en poursuivre l'étude). C'est donc d'une approche pavée de petits échecs, d'avance tatillon et peu rapidement que j'ai pu amener des systèmes solutions viables.

Une difficulté à part entière à laquelle je me suis confronté est issue de mon manque d'expérience naturel en tant qu'étudiant. Que ce soit en lisant des articles ou en m'intéressant à des méthodes, les explications sont bien souvent destinées à des gens expérimentés, des chercheurs, et les portes d'entrées des domaines à l'état de l'art de la reconnaissance visuelle, par exemple, sont lourdes et fastidieuses à emprunter. Vous remarquerez alors peut être un aspect de vulgarisation scientifique de ma part dans certaines parties de mon état de l'art, trahissant ma volonté de créer une porte d'entrée plus légère pour les lecteurs (dont je ferais partie si le temps me fait oublier certains détails).

Enfin et d'un point de vue purement technique cette fois-ci, j'ai été amené à utiliser des technologies parfois mal documentées et difficile à utiliser autrement que dans un cadre expérimental. J'ai aussi été confronté au déploiement d'un système d'imagerie sur un serveur isolé de tout composant graphique. Lors de ce stage, beaucoup de temps a été consumé par des problèmes bloquants simples mais difficiles à contourner car issus de particularités techniques non explicite ou non indiquées. C'est donc sans surprise qu'une des difficultés du stage ait été de supporter l'aspect irrégulier de ma progression, là où réussir à implémenter une solution viable est plaisant, échouer à résoudre rapidement des problèmes d'ordre technique est naturellement difficile.

Pour conclure, j'ai rencontré beaucoup de difficultés au cours de mon stage mais rétrospectivement, le fait de m'y être confronté et de les avoirs résolu, pour la plupart, constitue une connaissance qui me permet de classer ces mauvaises expériences en tant qu'acquis.

5.2 Acquis

Les acquis résultants de ce stage sont conséquents. Pour commencer le temps passé au sein de l'entreprise et de ses employés a contribué à me constituer une meilleure vision de la vie en entreprise et sur les rouages qu'il s'y tient. Ensuite, les parties de recherche et rédaction de l'état de l'art m'ont permis de renforcer mes connaissances et ma force de proposition en matière de système solution dans les domaines du traitement de l'image et de la vision par ordinateur ; ce qui me sera très utile par la suite. Le stage m'a aussi permis de découvrir la surface d'autres domaines que ce soit en observant les présentations des travaux de mes collègues ou en déployant une solution dans un contexte non prévu initialement (web-service). Ainsi, j'ai gagné beaucoup d'expérience et d'affinité avec les outils de visions par ordinateurs incontournables et j'ai surtout pu en découvrir de nouveaux. Enfin, d'un point de vue méthodique, je pense maintenant être beaucoup plus apte à élaborer des systèmes solutions rapidement pour des problématiques autour de l'imagerie.

Pour conclure, j'ai appris et gagné au-delà de ce que je pouvais attendre d'un stage de six mois et je suis fière du travail fourni qui forme et consolide, avec mes études, une base sur laquelle je vais pouvoir m'appuyer.

5.3 Synthèse : Retour sur expérience

Par rétrospective, le stage m'a permis d'être de rencontrer plusieurs profils intéressants travaillant sur des problèmes différents : des ingénieurs, des doctorants, des docteurs, des enseignants…Ce stage a été l'occasion de mettre en pratique certains aspects que j'avais vu durant mes études, mais aussi de me consacrer pleinement à un sujet donné sur une période de temps. Des travaux personnels antérieurs utilisant des méthodes, matériel et technologies similaires m'ont permis de gagner du temps sur certains aspects des projets et la nouvelle expérience acquise me permettra d'en gagner davantage. Ce stage me permet de m'orienter définitivement vers le traitement du signal et particulièrement celui de l'image après des études où le choix restait à faire (traitement des langues, du son …). En rédigeant ce rapport, je me suis aperçu que mes méthodes se sont améliorées en l'espace de six mois et que par

conséquent, mes algorithmes et mes choix sur les premiers projets peuvent être enrichi par les acquis des projets suivants.

J'aimerais aussi indiquer que beaucoup de choses ne trouvent pas leurs places dans ce rapport, mais que la phase d'étude et développement m'a permis de réaliser des expérimentations et des implémentations très formatrices dans le cadre des études des solutions. Je pense notamment aux tentative d'élaboration d'une architecture d'un réseau de neurones convolutif simple (Caffe) et la découvertes d'outils de reconstruction 3D d'environnements basés sur des points d'intérêts (RTabMap) entre autres…

Pour finir, je suis ravi d'avoir eu l'opportunité de travailler sur les problématiques d'imageries de Cordon DS2I avec les employés du site de Val-de-Reuil. Fort de cette expérience, je compte bien continuer de mettre à l'épreuve mes compétences et mes nouvelles connaissances afin de continuer de me spécialiser dans ce domaine qui suscite tant mon intérêt : le traitement de l'image.

Glossaire

blocs Les blocs (ou "blocks") sont les sous-matrices d'une image, des régions définies par une largeur et une hauteur, souvent égales. Dans le cadre de ce stage, les blocs choisis sont de tailles 8x8, 16x16 ou 32x32. 15

bottleneck De l'anglais "goulot". Terme désignant l'étape dans l'entraînement d'un réseau de neurones précédent celle qui détermine les caractéristiques de références pour la classification. 45

C++ Langage de programmation compilé permettant la programmation sous de multiples paradigmes comme la programmation procédurale, orientée objet et générique. 40

CVPR Computer Vision and Pattern Recognition. 6

Deep Learning De l'anglais "Apprentissage Profond". Ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. 19

DS2I Défense Sécurité Imagerie & Innovation. 1,

framework De l'anglais "structure", "système" ou "cadre". En programmation informatique, un framework est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel. 41, 43

HDMI De l'anglais "High-Definition Multimedia Interface". Norme et interface audio/-vidéo totalement numérique pour transmettre des flux chiffrés au maximum et non à minima, généralement non compressés et destinée au marché grand public. 6

HSV Hue Saturation Value (espace colorimétrique). 7

inférence Opération par laquelle on passe d'une assertion considérée comme vraie à une autre assertion au moyen d'un système de règles qui rend cette deuxième assertion également vraie. 19

Inpainting Méthode de réparation d'images basée sur la labelisation des régions à reconstruire. En s'inspirant du voisinage de la région marquée, celle-ci recréer la zone manquante ou détériorée. 27

LPIRC Low-Power Image Recognition Challenge. 6

LSVRC Large Scale Visual Recognition Challenge. 6

Matching De l'anglais : "correspondre", "correspondance". Utilisé par abus de langage pour désigner le processus de mis en relation de Pattern avec une Scène. 11

Open-Source De l'anglais "Sources libres/gratuites/ouvertes". Les projets open-sources sont des projets dont la licence est considérée comme libre. Plus d'information sur `https://opensource.org/` . 42, 43

OpenCV OpenCV (pour "Open source Computer Vision") est une bibliothèque libre fournissant l'architecture nécessaire au développement de solutions avec un système de vision par ordinateur. Elle permet, à l'aide de plus de 2500 algorithmes optimisés disponibles et de sa communauté de 47.000 utilisateurs, de faciliter ainsi que d'accélérer le travaille des chercheurs et des ingénieurs.
Voir `http://opencv.org/about.html` pour plus d'informations. 40

Pandore Pandore est une bibliothèque standardisée d'opérateurs de traitement d'images. 41

Pattern De l'anglais : "Modèle". Utilisé par abus de langage pour désigner des images de références, souvent petites et nécessaire à la correspondance de celle-ci dans une scène plus grande. , 50, 51

PCL Point Cloud Library.

Protobuf Protobuf, de l'anglais "Protocol buffers", est un format de sérialisation avec un langage de description d'interface développé par Google. 41

RANSAC de l'anglais "**RAN**dom **SA**mple **C**onsensus". Dans notre contexte : Méthode pour estimer les paramètres de certains modèles mathématiques (dans notre cas, des plans) tout en ignorant certaines valeurs aberrantes. 25

RECOB Pour "Reconnaissance d'objets". Le premier sujet et projet du stage. 5

RGBD Red Green Blue Depth. 42

RJ11 Registered Jack standard 11. Nom usuel du connecteur utilisé pour les appareils téléphoniques fixes. La prise RJ11 est plus petite et dispose de moins de fils qu'une prise RJ45. 6

RJ45 Registered Jack standard 45. Nom usuel du connecteur 8P8C (8 positions et 8 contacts électriques) utilisé couramment pour les connexions Ethernet, et plus rarement pour les réseaux téléphoniques. 6

ROI Region of Interest (Région d'intérêt). 11,

ROS Robot Operating System. 41

Scène La scène décrit ici l'espace d'acquisition de l'image et est composée de tous les objets présent dans le champs de vision de la caméra. , 50

SERVED Structure Extraction for Recognition of Visual Elements in Documents. 31

Template Voir Pattern.

TFD Transformée de Fourier Discrète. 38

Transfer Learning De l'anglais "Apprentissage transféré". Le partage de réseau de neurones pré-entraîné pour une utilisation détournée ou pour en ajuster la dernière couche. Cette possibilité propose l'avantages d'un gain de temps considérable pour des résultats à hauteur de l'état de l'art. 19, 23

Vision par ordinateur La vision par ordinateur (aussi appelée vision artificielle ou vision numérique) est une branche de l'intelligence artificielle dont le principal but est de permettre à une machine d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition (par exemple : caméras, etc.). 3, 5

VOC (Pascal) Visual Object Classes. 6

Watershed De l'anglais "ligne de partage des eaux". Méthode de segmentation qui consiste à traiter une image comme un relief où la progression dans la $3^{\text{ème}}$ dimension est représentée par le gradient d'intensité. On "inonde" volontairement cette représentation avec des points de source d'eau qui se propage uniquement vers le bas (gradient moins élevé) afin d'observer une propagation. La segmentation a lieux entre les régions émergées et inondées. 29

Références

[1] Creating a set of training images. URL : https://www.tensorflow.org/versions/master/how_tos/image_retraining/index.html#creating-a-set-of-training-images.

[2] IPOL journal · non-local means denoising. URL : http://www.ipol.im/pub/art/2011/bcm_nlm/.

[3] OpenCV – bounding box & skew angle | félix abecassis. URL : https://felix.abecassis.me/2011/10/opencv-bounding-box-skew-angle/.

[4] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. 13(2) :111–122.

[5] Serge Beucher and Fernand Meyer. The morphological approach to segmentation : the watershed transformation. *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 34 :433–433, 1992.

[6] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6) :679–698, 1986.

[7] Hongli Deng, Wei Zhang, Eric Mortensen, Thomas Dietterich, and Linda Shapiro. Principal curvature-based region detector for object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[8] Lei Ding and Alper Yilmaz. Enhancing interactive image segmentation with automatic label set augmentation. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010.

[9] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1) :11–15, 1972.

[10] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge : A retrospective. 111(1) :98–136.

[11] Neil Fraser. Neural network follies, 1998. URL : https://neil.fraser.name/writing/tank/.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL : http://arxiv.org/abs/1512.03385.

[13] Nawal Houhou, Xavier Bresson, Arthur Szlam, Tony F Chan, and Jean-Philippe Thiran. Semi-supervised segmentation based on non-local continuous min-cut. In *Scale Space and Variational Methods in Computer Vision*, pages 112–123. Springer, 2009.

[14] John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1) :87–116, 1988.

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv :1607.02533*, 2016.

[16] Hongyang Li, Huchuan Lu, Zhe Lin, Xiaohui Shen, and Brian Price. Inner and inter label propagation : Salient object detection in the wild. *Image Processing, IEEE Transactions on*, 24(10) :3176–3186, 2015.

[17] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 89–96. IEEE, 2011.

[18] Esther Nicart, Bruno Zanuttini, Hugo Gilbert, Bruno Grilhères, and Fredéric Praca. Building document treatment chains using reinforcement learning and intuitive feedback. JFPDA, France, July 2016. URL : `http://jfpda2016.imag.fr/lib/exe/fetch.php/jfpda_2016_paper_02.pdf`.

[19] Sarbajit Pal, Pankaj Ganguly, and PK Biswas. Cubic bézier approximation of a digitized curve. *Pattern recognition*, 40(10) :2730–2741, 2007.

[20] John W Peterson. Arc length parameterization of spline curves. *Journal of Compu ter Aided Design*, 2006.

[21] B Smith and R Gosine. Support vector machines for object recognition, 2001.

[22] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1) :32–46, 1985.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. URL : `http://arxiv.org/abs/1512.00567`.

[24] Fei Wang, Xin Wang, and Tao Li. Efficient label propagation for interactive image segmentation. In *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pages 136–141. IEEE, 2007.

[25] Lei Zhang, Xun Wang, Nicholas Penwarden, and Qiang Ji. An image segmentation framework based on patch segmentation fusion. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 187–190. Ieee, 2006.

[26] Song Chun Zhu and Alan L Yuille. Forms : a flexible object recognition and modelling system. *International Journal of Computer Vision*, 20(3) :187–212, 1996.

Table des annexes

État de l'art
Reconnaissance visuelle automatique d'objets

Valentin Laforge

31 août 2016

Table des matières

1 Vision par ordinateur : reconnaissance d'objets

1.1 Reconnaissance générale

En vision par ordinateur, un système de reconnaissance visuelle d'objets établis une **distance de similarité** entre un **objet de référence** (ou les éléments qui le caractérisent) et les éléments d'une **scène**. Cette dernière peut être complexe, encombrée, parasitée ; on cherche toutefois à vérifier la présence d'une instance de la référence dans celle-ci. Pour cela, il convient d'établir un ensemble venant décrire l'objet (descripteurs) et une méthode de calcul de similarité.

Décrire un objet dans une matrice de pixels en deux dimensions peut être réalisé à plusieurs niveaux. Il est commun de décrire l'objet par ses **points d'intérêts** qui sont des régions de l'image avec des propriétés caractéristiques (il s'agit souvent de contours, de bords, de trous etc. Cela varie selon la méthode utilisée). Il est aussi possible d'utiliser les **informations colorimétriques** d'un objet, sa **forme** ou même d'approcher la problématique en travaillant sur ses **textures**. Enfin, des modèles plus abstraits (plus expérimentaux et moins précis) peuvent être utilisés, comme la réalisation de la moyenne de la scène attendue (la reconnaissance aurait alors lieue au niveau de la scène avec ses objets et non plus de l'objet même) ou même en utilisant d'autres types d'ondes électromagnétique que la lumière.

Travailler en utilisant la **profondeur (spatiale)** permet d'obtenir une description plus fine de la référence ou de la scène. La reconnaissance est alors basée sur une interprétation géométrique de l'objet dans l'espace. Elle peut aussi servir à des fins de segmentations, pour mieux détecter les contours à extraire d'une image.

Une quatrième dimension (temporelle) est susceptible de favoriser l'établissement d'une homographie. La reconnaissance se réalise alors sur un **flux composé d'images séquentielles**. Le but est d'obtenir la ou les correspondance(s) à un moment t de la séquence. Prolonger cette correspondance sur plusieurs instants d'une suite d'images permet de traquer l'objet dans l'espace, on parle alors de "**tracking**". Selon le type de séquence (courte, longue, en boucle), le calcul de similarité peut être **ponctuel** ou **continu**.

L'**effort** de reconnaissance est réalisé par un ou plusieurs processeurs (graphique ou non) aidé de mémoire de stockage morte et vive. L'opération possède donc un **coût de traitement en temps** qui est directement lié à la capacité de l'unité de calcul et le ou les algorithmes utilisé(s). Chercher à réduire ce coût revient à agir sur ces deux facteurs. Deux délais supplémentaires d'un système de reconnaissance sont à prendre en considération, il s'agit du temps d'acquisition par le capteur (souvent négligeable) et, dans le cas où le système est supervisé, du temps d'utilisation du système par l'homme. Cela nous mène alors sur deux orientations méthodiques : le traitement en temps réel ou différé.

Un système de reconnaissance en temps réel est en pratique un système de reconnaissance effectuée en temps de calcul ajouté au temps de début d'opération. Si le temps de traitement est acceptable pour l'homme, on estimera que l'opération se déroule en **temps réel** et on peut chercher à avoir un résultat immédiat. Si au contraire le temps de traitement est long, on parlera de reconnaissance en **temps différé**. Dans ce second cas, le système de vision et d'évaluation peuvent être séparés et traiter les cas à différentes vitesses.

1.2 Connaissance

Tout système fonctionnel de reconnaissance nécessite une **base de connaissances**. L'ampleur de celle ci peut varier d'un simple code couleur à une large base d'images. Certains systèmes sont capables de classifier l'information par l'**apprentissage**, en apprenant depuis une banque d'images labellisées ou même sans utiliser de données supplémentaires. Dans ce dernier cas, ils se constitueront eux-mêmes leurs connaissances (ils reconnaîtrons des objets similaires sans pour autant savoir de quoi il s'agit). Reconnaître un objet, c'est donc avoir une attente d'un certain nombre de **caractéristiques** de la part d'une scène. Ces caractéristiques peuvent être locales, globales et abstraites si isolées. On préférera travailler avec des propriétés **intrinsèques** (invariantes à la rotation, par exemple), **dense** (très caractéristique) et **compactes**. La taille de la base de connaissance ainsi que son contenu peut alors varier ; on peut avoir une base de connaissance décrivant un objet avec des points clés, ses contours, sa modélisation en 3D etc. Même si, en pratique et en vision par ordinateur, on se base souvent sur une photo de l'objet qui sera traitée plus tard (par soucis de robustesse à l'utilisation de n'importe quelle photo), il reste possible de stocker durablement uniquement des descripteurs (on perd alors en flexibilité pour gagner en taille de

stockage de l'information).

La reconnaissance se base sur la connaissance, néanmoins, elle n'a pas pour but de retrouver ce qui constitue la connaissance : elle a pour but d'identifier des similarités entre les nouvelles informations à traiter et les anciennes, déjà classifiées. Cette notion est importante : aucun système de vision intelligente, qu'il soit humain ou informatique, ne peut fonctionner efficacement sans accepter de différences entre la référence et l'objet à identifier. Ainsi, **la connaissance n'est pas nécessairement exhaustive** (connaissance de l'objet sous tous les angles, toute luminosité ...).

1.3 Acquisition

Pour l'homme comme pour la machine, reconnaître passe avant tout par l'étude d'un cas dont il ou elle doit prendre connaissance. Pour les deux, nous venons de voir que la distinction se fait généralement sur des critères abstraits et non nécessairement exhaustifs. En effet, la vue très précise de l'homme permet d'acquérir la couleur et la profondeur mais il n'utilise que très peu cet avantage et construit sa connaissance depuis les éléments les plus caractéristiques de sa vision. Pour les machines, ce sont les capteurs qui accordent la vue au système. Ainsi, et suivant le capteur utilisé, la vision de la machine peut différer de celle de l'homme.

1.3.1 L'œil

L'organe humain de la vue vient par pair ; il est photosensible, permet la vision stéréoscopique et l'interprétation de la profondeur. L'adaptation à la luminosité est réalisée grâce à un muscle, l'iris (partie colorée de l'œil) qui obstrue la pupille en fonction de la quantité de lumière qu'il souhaite laisser passer. Les rayons de lumières admis passent alors par le cristallin dont la forme de lentille permet de focaliser les rayons lumineux vers la rétine. Ce capteur transmettra alors l'acquisition au cerveau par le nerf optique pour traitement.

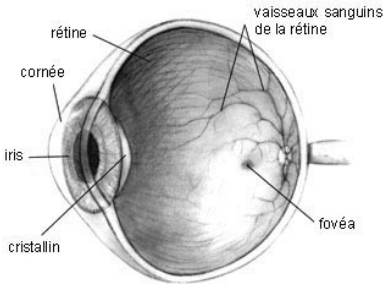


FIGURE 1 – Dessin en coupe de l'œil

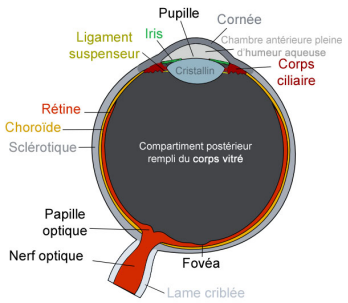


FIGURE 2 – Schéma en coupe de l'œil

Les cônes et les bâtonnets de l'œil sont les instruments photosensibles du capteur. Les bâtonnets sont sensibles à tout type de lumière quand les cônes sont sensibles aux trois couleurs primaires : Rouge, Vert et Bleu. Des neurones lies au cerveau chacun des bâtonnets et des cônes dont la stimulation par la lumière envoie une impulsion qui sera traitée par ce second grâce à un procédé complexe encore mal connu à ce jour.

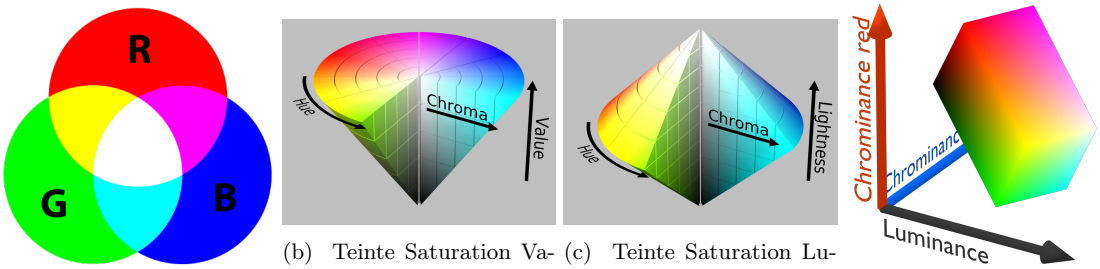
Nous pouvons aussi retenir deux caractéristiques intéressantes de l'œil. La première est que les cellules de la périphérie de l'œil sont très sensibles au mouvement, ce qui permet d'acquérir une présence et/ou un mouvement sans réellement le voir. La seconde est que les instruments photosensibles dont nous avons parlés sont inégalement réparties dans l'œil. En effet, ceux-ci se concentrent autour de la fovéa (fond de l'œil, diamétralement opposé à la rétine), ayant pour effet de créer un "focus" dans notre vision, c'est-à-dire une zone très détaillées en comparaison à son entourage, qui devient flou. La précision des yeux est ainsi liée à la capacité de l'homme à aligner sa rétine en direction d'un même point.

1.3.2 Capture d'images 2D

Pour effectuer notre reconnaissance, nous avons besoin d'un capteur afin d'acquérir la scène contenant le ou les objet(s) à reconnaître. Dans cette partie, nous parlerons des outils photosensibles afin de traiter

des images restituées en deux dimensions, dans une matrice de pixels. Nous parlons alors d'une caméra pouvant acquérir une image ou une séquence d'images. De nos jours, ce genre de système d'acquisition est très accessible et abordable. En matière de reconnaissance, la nécessité du détail est restreinte, rendant pertinent l'utilisation des webcams les moins performantes.

L'information acquise est encodée afin de pouvoir être temporairement stockée. Le format le plus utilisé et semblable à l'homme est celui qui utilise les couleurs primaires à savoir le rouge, le vert et le bleu (on parle alors de RVB ou RGB, en anglais). La matrice de l'image admet alors en tout pixel les valeurs de trois canaux, un pour chaque couleur primaire. D'un point de vue algorithmique et pour faciliter certains traitements, il est judicieux de représenter l'image par d'autres méthodes de codages tels que HSV (de l'anglais "Hue Saturation Value", teinte saturation valeur) ou encore HSL (de l'anglais "Hue Saturation Lightness", teinte saturation luminosité) parmi d'autres (YCbCr, "Luminance ; Chroma Blue ; Chroma Red"...) pour mieux traiter l'information.



(a) Rouge Vert Bleu (b) Teinte Saturation Valeur (c) Teinte Saturation Luminosité (d) YCbCr

FIGURE 3 – Représentations de quelques espaces colorimétriques

1.3.3 Capture d'images avec profondeur 3D

Nous parlerons de capture en trois dimensions lorsque la notion de profondeur sera acquise. Celle-ci peut étendre les captures en deux dimensions vues précédemment et s'intégrer en tant que quatrième canal de la représentation de l'image. Pour ce type d'acquisition, les outils doivent donc être adaptés. Trois possibilités s'offrent au système :

1. Utilisation de **deux caméras en champs croisés** ou d'une caméra **stéréoscopique** [43]. La méthode de visualisation est la même que celle utilisée par les yeux de l'homme.
2. Utilisation d'un **capteur de profondeur** pour une restitution 3D de la scène [102].
3. Mise en correspondance d'objets 3D avec soit une image 2D, soit une estimation 3D de la scène depuis une image 2D ([19], [27], [75]). Ce cas particuliers n'impose pas l'utilisation de plus d'une caméra "normale" mais amène une grande part d'estimation et de calcul à la reconnaissance d'objets.

Ces méthodes de visualisation restent **incomplètes** : même si les données décrivent plus fidèlement l'objet dans la scène, une partie de l'information sur celui-ci reste inaccessible (faces cachées ou occultées par les autres objets de la scène). Seule une méthode avec un plus grand nombre d'angles d'acquisitions (une prolongation des méthodes 1 et 2 avec soit un plus grand nombre de caméras [109] [91], soit une opération de rotation) permettrait d'établir une description complète. Ici, la performance des temps de traitement(s) ainsi que les moyens de calculs et de manipulation(s) associés sont considérablement sanctionnés au profit d'une précision accrue grâce à un ensemble de description plus complet des objets à reconnaître.

1.4 Méthodes de reconnaissance

1.4.1 Similarité depuis des points d'intérêts détectés et décrits

La **détection** et la **description** des points d'intérêts peut être réalisée à l'aide de différents algorithmes. Les algorithmes de détection se concentrent principalement sur la localisation de features, de zones caractéristiques de préférences intrinsèques avec une forte teneur en informations (les coins des objets présents dans la scène entre autres [39]). Les algorithmes de description sont utilisés pour décrire

ces zones/points. Certaines méthodes proposent des détecteurs (aussi appelés extracteurs) avec des descripteurs, d'autres non. Ce n'est pas un soucis que d'utiliser une méthode de description différente de la méthode de détection tant qu'elles sont compatibles. Les caractéristiques extraites peuvent aussi être traitées comme du texte, on utilisera alors un vocabulaire visuel [55] tel que VLAD [23] (qui donne de meilleurs résultats que BOW). On cherchera alors à effectuer la reconnaissance en réalisant des distances sur les descriptions entre les points d'intérêts détectés sur l'objet et la scène. Une faible distance entre deux points indique une similarité locale ; une faible distance entre deux ensembles de points sur une région indique une similarité au niveau de l'objet. Grâce à ces méthodes, l'objet peut être **classifié** et **localisé**. Parmi les propositions de reconnaissance visuelle d'objets par points d'intérêts, existent :

1. **SIFT** [67] (*Scale-invariant feature transform*)
2. **SURF** [7] (*Speeded-Up Robot features*)

Ces deux algorithmes précédents sont néanmoins brevetés aux États-Unis. Bien qu'ils soient les méthodes les plus connus pour la détection et l'extraction de features locales, il existe des alternatives viables et parfois même plus robustes. Sélectionner SIFT ou SURF hors cas d'études pour la reconnaissance d'objets ne semble donc pas pertinent. Depuis, des alternatives non brevetées ont été proposées, proposant parfois de meilleurs résultats :

1. **ORB** [92] (*Oriented FAST and Rotated BRIEF*)
2. **KAZE** ([5], [18]) (*"vent" en Japonais*)
3. **A-KAZE** [82] (*Accelerated-KAZE*)
4. **BRIEF** [15] (*Binary Robust Independent Elementary Features*)
5. **BRISK** [63] (*Binary Robust Invariant Scalable Keypoints*)
6. **FAST** [113] (*Feature from Accelerated Segment Test*)
7. **FREAK** [4] (*Fast Retina Keypoints*)
8. **GLOH** [81] (*Gradient Location and Orientation Histogram*)
9. **HOG** / **PHOG** / **DOG** / **LOG** (*Histogram/Pyramidal Histogram/Difference/Laplacian of Gaussian*)
10. **LDB** [117] (*local Difference Binary*)
11. **LIOP** [115] (*Local Intensity Order Pattern for feature description*)
12. **LBP** [42] (*Local Binary Pattern*)

Les différents articles et comparatifs montrent que la liste des descripteurs sont viables. Toutefois, certains sont plus robustes à :
— la translation,
— la rotation,
— l'échelle,
— l'occultation,
— au bruit.

On retiendra alors les méthodes suivantes parmi les plus performantes (basé sur les critères de temps d'exécution, précision, rappel) :
— BRIEF,
— BRISK,
— KAZE / A-KAZE,
— ORB.

Attention, la composition des différents comparatifs présents dans les articles ne nous permet pas d'avoir un point de vue optimal pour déterminer le meilleur descripteur. L'idéal serait d'implémenter, paramétrer et comparer ces méthodes sur une même base d'images (comportant les objets qui nous intéressent pour un système de reconnaissance) et le même matériel. Les précisions étant presque identiques, nous pouvons alors sélectionner sans trop de craintes une méthode d'extraction des points d'intérêts parmi les 4 précédentes tout en ayant de bons résultats dans le domaine.

1.4.2 Apprentissage automatique et interactif

L'utilisation de machines à vecteur de support [11] est aussi possible pour reconnaître des ensembles similaires par région [116] ou l'extraction d'objets dans les images [61, 58]. Ce moyen de classification est souvent utilisé pour résoudre des problèmes de discrimination ou de régression, il peut donc s'avérer efficace pour classifier des objets [99, 69]. Ils sont notamment très utiles pour établir une reconnaissance intra-classe sur les groupes génériques [59]. En effet, l'objectif des méthodes dont nous allons maintenant parler est l'identification de clusters dans un plan pour catégorisation. Elles se résument donc à établir des frontières de séparations entre les coordonnées des points caractéristiques dans le plan. La méthode de clustering est alors à déterminer suivant la problématique (coût de calcul, complexité de la segmentation des clusters et tendance à s'attacher aux données). Les méthodes de classifications (listé de façon représentative et non exhaustive) peuvent alors être les suivantes :

- **K-moyennes**, qui repose sur le fait que les clusters soient convexes et très identifiables,
- Modèle de mélange de gaussienne, qui propose une meilleure classification pour les clusters plus complexes mais toujours convexes,
- **Clustering spectral**, qui nous permettrai de classifier les données non convexes, dont la représentation des clusters est non sphérique,
- **SVM linéaire ou polynomial multi-classe**,
- **RandomForest**, qui fonctionne par apprentissage depuis des arbres de décisions entraîné sur des ensembles de données peu variants,
- **AdaBoost**, classificateur binaire basé sur la notion de Boosting utilisant des règles simples.

Bien qu'ils approchent la reconnaissance d'objet par l'apprentissage, ces méthodes peines à proposer des solutions à hauteur de l'état de l'art. Ils sont néanmoins utiles lors d'étapes intermédiaires [112] de propositions de reconnaissance et dans des cas très particuliers. Plus d'informations sur l'apprentissage interactif appliqué aux images ainsi que la notion de boosting sont disponibles dans la thèse sur l'"Apprentissage interactif et multi-classes pour la détection de concepts sémantiques dans des données multimédia" d'A. Lechervy [60].

1.4.3 Réseaux de neurones convolutifs

Les méthodes les plus récentes et efficaces lorsqu'il s'agit de faire de la reconnaissance sur de larges bases d'images utilisent des **réseaux de neurones** [106, 22]. Ceux-ci s'inspirent du cerveau biologique et s'inscrivent dans la catégorie des perceptrons multi-couches. Grâce à une simulation logicielle d'un système neuronal, ils apprennent à reconnaître des éléments suite à un entraînement sur un jeu de données correspondant. L'apprentissage peut être supervisé ou non. Un apprentissage **supervisé** permettra au réseau d'"apprendre" en connaissance des sorties attendues (donc des éléments à reconnaître), il nécessite donc un jeu de données déjà identifiées (données labelisées avec la vérité terrain). Un apprentissage **non supervisé** laissera le réseau évoluer à l'aveugle pour reconnaître des classes objets qu'il ne saura nommer à l'homme ; ce second cas est utile dans les systèmes de reconnaissances "autonomes" [64] (un compagnon robotisé à la découverte de son environnement, par exemple) quand le premier (supervisé) est plus utile dans les systèmes de reconnaissances restreintes et finies. La taille et la pertinence de la base d'apprentissage influe sur les résultats ; plus elle est importante et bien conçue (sur de bons cas d'études), plus elle fait converger la précision d'un réseau de neurones vers sa capacité de reconnaissance optimale (de façon non linéaire). L'apprentissage prend fin quand on estime que le réseau a finis de converger vers sa précision optimale.

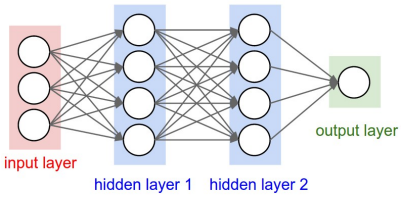


FIGURE 4 – structure d'un réseau de neurones

En vision par ordinateur, les réseaux de neurones **convolutifs** font partie des systèmes de reconnaissance(s) les plus complexes ([30, 46]) et les plus intéressants. En effet leurs neurones offrent les capacités perceptives en se basant sur des filtres de convolutions aux noyaux et décalages différents afin de mettre

en avant les caractéristiques les plus descriptives du contenu d'une image (selon le filtre). On décale donc un masque de convolution sur une image en cherchant à prendre en compte le voisinage ; ainsi la taille de ces masques est généralement de 3×3 pixels. Dans la plupart des cas et dans les réseaux à l'état de l'art on affine les résultats en réalisant plusieurs convolutions différentes pour un même cas d'étude. Cela permet de mettre en avant des caractéristiques différentes pour l'apprentissage ; on obtient alors un réseau d'aspect multi-couches.

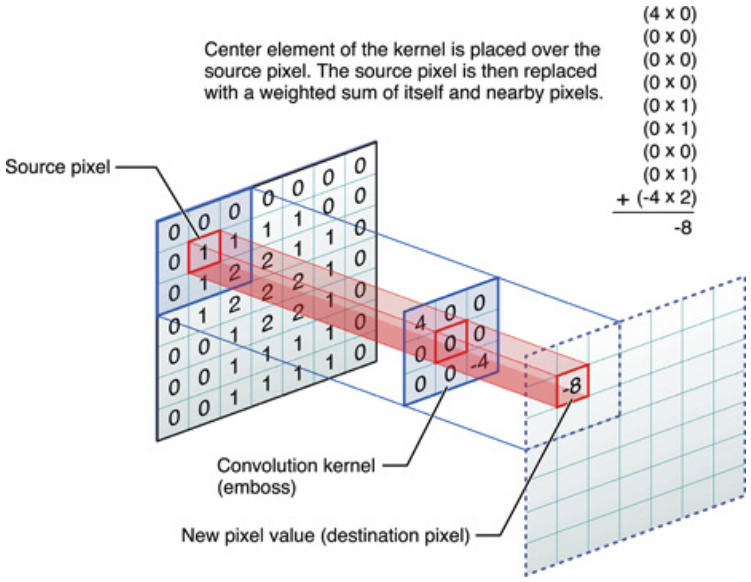


FIGURE 5 – Convolution avec un masque de taille 3×3

Les **neurones** d'un réseau à convolutions prennent des entrées généralement associées à des **poids initialisés aléatoirement**. Après avoir appliqué l'opération demandée, ils réagissent à une fonction d'activation binaire qui servira à déterminer la similarité. On choisit généralement d'améliorer l'activation en l'"adoucissant" par une autre fonction. On utilise alors généralement des fonctions **sigmoidal** ou des **tangentes hyperboliques** qui permettent de stimuler un peu plus le neurone ou au contraire d'inhiber le signal sans le supprimer (on peut ainsi éviter une sortie binaire du neurone). Cependant, ces fonctions ont tendances à faire disparaître le gradient lorsqu'elles atteignent des valeurs aux extrémités (0, 1). On utilise donc, plus récemment, des rectifications linéaires qui n'ont pas tendance à faire disparaître le gradient issu des filtres de convolutions et possèdent l'avantage de faire converger plus rapidement. Les sorties et les choix discriminants de chaque neurones sont donc abstraits ; nous pouvons toutefois appréhender l'apprentissage du réseau en gardant à l'esprit que les noyaux utilisés mettent généralement en avant des contours et des coins. Ainsi, chaque neurone répond à la fonction suivante $y = f(< w, x >) + b$ où w représente le poids et b la valeur venant biaiser la sortie du neurone.

La **classification** au sein d'un réseau convolutif a généralement lieu grâce à un procédé en trois étapes : appliquer un filtre de convolution, lui associer une fonction d'activation puis échantillonner les résultats selon un vote (softmax par exemple). La première est la partie qui imite les capacités de reconnaissance biologique quand la troisième sert à déterminer quelles caractéristiques du signal reçu servirons à la classification. L'architecture Inception-v3 [107] en propose un très bon modèle ; elle utilise un réseau de neurones convolutif capable de traiter des régions de l'image qui se chevauchent tout en réduisant les paramètres et les coûts de calculs nécessaires. Elle a dernièrement intégrée les notions de "Batch normalization"[47] et de réseau résiduel [40] pour évoluer sous le nom d'Inception-v4 [105] et converge plus rapidement. Ces solutions à la reconnaissance d'éléments d'une image sont au cœur de l'état de l'art à ce jour [41].

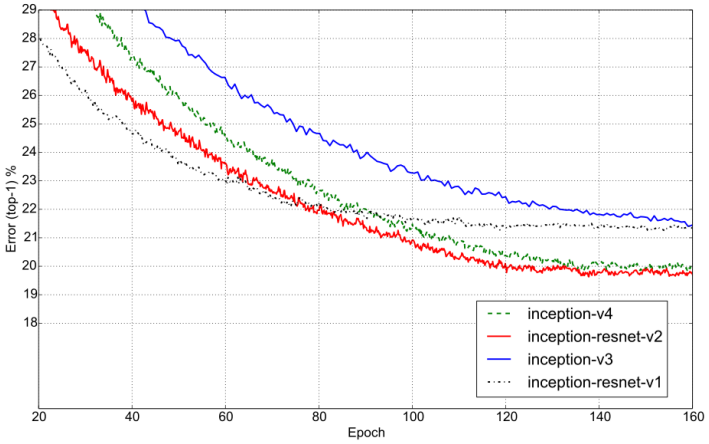


FIGURE 6 – Comparatif extrait de [105] illustrant les gains sur la convergence de la Batch Normalization et de la capacité résiduel des RDN sur une architecture convolutive déjà performante

Les réseaux de neurones appliqués à la vision par ordinateur prennent généralement en entrées des **images matricielles à taille réduite**. Comme nous l'avons vu, ils calculent des **goulots** qui seront utilisés pour représenter le modèle de la classe étudiée. Ceux-ci, ainsi que leur criticité, sont directement dépendants de la nature des données traitées. Dans le cas où nous avons plusieurs canaux (un par couleur primaire, par exemple) il est possible de les passer en entrée un à uns dans plusieurs réseaux de neurones convolutifs dédiés [6], afin de calculer les goulots suivant la correspondance colorimétrique de l'élément. Les résultats sont alors combinés par vote majoritaire ou par moyenne. Ainsi, la taille des matrices de pixels a tendance à diminuer au cours du traitement. Afin d'éviter cela pour, par exemple, répondre à des problèmes de segmentation, il est possible de changer l'architecture du réseau en faveur d'un réseau "fully convolutional" [98] (entendez "entièrement convolutifs", même si cela n'est pas réellement le cas). De même, la profondeur d'une image n'étant pas nécessairement représentée de façon visuelle, elle peut être traitée par un réseau à l'architecture différente, par exemple, un réseau de neurones récursif [101]. Il existe aussi, notamment dans des problématiques de reconnaissances textuelles, des RDN à capacité mémoriel. On parle alors de réseau de neurones récurrents qui possèdent la particularité de parcourir une image par bloque en conservant les informations du bloque au temps t-1 [84]. Une autre façon d'utiliser une notion de mémoire avec les RDN et de les mettre en cascade [103]. Cela consiste à réaliser un réseau de réseaux de neurones depuis les différentes étapes d'apprentissages d'un ou plusieurs RDN sur une même problématique. Enfin, on peut utiliser un ensemble de réseaux de neurones différents pour étudier, par exemple, un cas acquis en RGBD (Red Green Blue Depth) [6] ; on cherche alors et dans ce cas à utiliser un RDN par couche d'information (ici 4 : rouge, vert, bleu et profondeur). Une dernière application des réseaux de neurones convolutifs à l'espace géométrique consiste à réaliser plusieurs reconnaissances d'un même ensemble sous plusieurs angles de vues, on parle alors de reconnaissance 'Multi-view' [87].

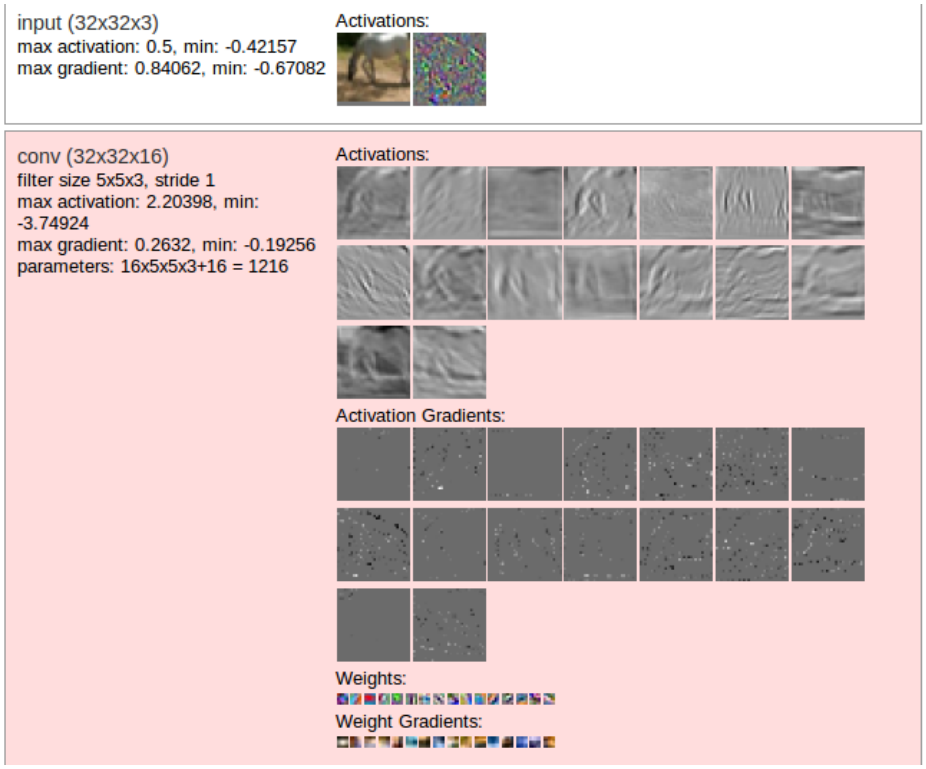


FIGURE 7 – Exemple (extrait de ConvNetJs) de ce que peut mettre en évidence un neurone sur les premières couches cachées d'un RDNC sur un cas simple (CIFAR-10 images de taille 32x32)

La nature laborieuse de l'entraînement des réseaux de neurones a introduit la notion de **transfer learning**. Le principe qui se cache derrière l'"apprentissage transféré" est tout simplement le partage de la partie paramétrable d'un réseau entraîné. En effet, en ajustant uniquement la dernière couche déterministe d'un réseau déjà paramétré sur une tâche similaire (traitement d'images, de textes ou autre), on obtient des résultats au niveau de l'état de l'art sans avoir à paramétrer et entraîner entièrement un nouveau réseau. De plus, la diversité entre les classes qui ont précédemment entraînées le réseau sur un problème plus grand et les nouvelles qu'on lui introduit lui font gagner en précision. Cependant, ajuster consécutivement et de façon répétée la dernière couche d'un réseau de neurones pré-entraîné sur une large base d'images (celle ILSVRC2012, par exemple) sur un problème restreint tend à l'affaiblir. Ainsi, le transfer learning permet, en plus du gain de temps, de préserver la diversité nécessaire à l'optimalité des capacités d'ajustement d'un réseau pré-entraîné ; il est donc bénéfique pour la communauté d'encourager le partage un réseau pré-entraîné. La justification derrière les gains de l'apprentissage transféré semble résider dans le simple fait qu'il est préférable d'instancier les poids d'un réseau de neurones depuis un ajustement réalisé sur un problème existant que de façons aléatoire.

Les méthodes énoncées dans cette partie sont coûteuses à mettre en place au niveau du prix du matériel et/ou le temps d'**apprentissage** nécessaires. Les **outils utilisés** pour la conception logique de réseaux de neurones sont des ordinateurs ultra-performants, souvent dotés de clusters de cartes graphiques à la pointe de la technologie. Ils sont sollicités pendant plusieurs heures, jours, semaines ou mois afin de mener à bien l'apprentissage des classes étudiées. Une approche encore sous-exploitée consiste à concevoir le réseau de neurones d'un point de vue matériel, en utilisant une **puce cognitive** comme TrueNorth du projet IBM SyNAPSE (systèmes d'électronique plastique neuromorphiques Adaptatifs). Celles-ci sont des reconstructions technologiques de cerveaux, constituées de neurones physiques. Si elles n'ont pas encore les mêmes capacités caractéristiques que le cerveau humain, elles s'élèvent néanmoins au niveau des souris et leur dévouement aux systèmes d'apprentissages leurs permettent de consommer beaucoup moins d'énergie qu'un ordinateur classique. Plus d'informations sur l'une des puces sur : (`http://www.research.ibm.com/articles/brain-chip.shtml`).

1.4.4 Reconnaissance géométrique

Les méthodes de reconnaissances portent généralement sur des matrices de pixels d'intensités en deux dimensions, format très répandu et accessible (acquisition, coût de stockage etc.). Il est possible d'ajouter un troisième plan à l'espace de description afin de tenir compte de la profondeur. On peut alors parler de reconnaissance géométrique. Très utilisée en **robotique**, pour la création de compagnons, la perception qui en résulte permet au robot de comprendre l'environnement qui l'entoure et de reconnaître certains objets [110] [96] afin de les saisir, les éviter, etc.

On peut séparer les méthodes de reconnaissances géométriques en deux groupes : celles qui utilisent la géométrie de façons décomposée (avant la reconnaissance, en pré-traitement, ou en isolant certains aspects comme la profondeur) et celles qui savent reconnaître géométriquement les éléments.
Le premier groupe de reconnaissance est composé de méthodes qui utilisent des données géométriques pour ensuite approcher la reconnaissance par une autre voie. Ainsi, il est possible d'utiliser des objets décrits en trois dimensions pour extraire sur des plans 2D des images correspondantes à une vue. Elles seront ensuite utilisées par des méthodes de deep learning ou pour créer des signatures [77], on parle alors de reconnaissance "multi-view" [104] [87]. De même, on peut utiliser plusieurs réseaux de neurones pour traiter indépendamment chacun des calques d'une image RGBD [28] [101]. Le second groupe est constitué de méthodes travaillant directement sur la répartition géométrique des objets à reconnaître. Ainsi, certains réseaux de neurones réalisent un apprentissage sur des nuages de points [73] [100] pour reconnaître géométriquement des objets. Ceux-ci peuvent aussi (tout autant que d'autres méthodes) baser leurs reconnaissance sur des formes simples. On retrouve ainsi beaucoup de reconnaissance lignes (HOUGH), de plans(RANSAC) ou de formes plus détaillées (HOUGH généralisé) en robotique. Une autre approche consiste à extraire des descripteurs d'une scène 3D, comme par exemple les "points orientés" sur les objets [49].

De façon générale, la reconnaissance géométrique seule ne constitue pas une approche précise à la reconnaissance visuelle d'objets. En effet, elle ne permet pas d'approcher le cas d'étude que par sa forme basique qui n'est généralement pas très descriptive (impossible de faire la différence entre deux boites de céréales). Nous ne retiendrons donc pas la reconnaissance géométrique en tant que méthode viable à la reconnaissance visuelle d'objets, cependant, nous garderons à l'esprit que la profondeur permet souvent d'affiner de façon remarquable la plupart des méthodes de segmentations et au moins légèrement les méthodes fonctionnant par apprentissage.

Pour plus d'informations sur la reconnaissance géométrique, G.BUREL aborde le traitement de l'image par la reconnaissance d'objets 3D dans sa thèse ([13] chapitre 7 p.187).

1.4.5 Autres méthodes de reconnaissances

La reconnaissance visuelle d'objets consiste à segmenter un ou plusieurs objet(s) dans une image pour les identifier. La problématique concerne donc toute représentation retransmettant un signal sous forme d'imagerie numérique. Ainsi, la **thermographie** [10] [14] mène à des pistes sérieuses de classification. De même, une approche de perception par utilisation d'une onde **électromagnétique** différente [3] peut être considérée, notamment avec l'utilisation de certaines ondes comme le **Wi-Fi** [1] [2] [79] ; la partie imagerie est, avec cette technologie, restreinte voir inexistante [111].
D'autres systèmes de détection peuvent répondre à notre besoin. Bien qu'ils mèneraient plus difficilement à une bonne précision, ils peuvent se montrer solutions de cas particuliers (câbles, vêtements en tas...). On peut donc considérer l'utilisation de la **couleur** [68] ou la **texture** [95] pour la reconnaissance visuelle d'**objets complexes et variants**. Retrouver ce type d'objets peut aussi être effectué avec une approche **partielle** de ses caractéristiques [54], en se basant uniquement sur un ensemble qui ne le caractérise pas entièrement. Les **contours** [50] [32] [48] ou même la texture résultante du calcul de ceux-ci [75] pourrons indiquer la présence de ce type d'objets. Enfin, puisque ces méthodes ne sont que très rarement individuellement suffisantes à la reconnaissance multiple et précise, les **combiner** est une bonne façon d'améliorer la précision du système et de le rendre plus robuste à la **déformation** et l'**occultation** [89] [26].

1.5 Localisation et segmentation d'objet(s) détecté(s)

Une fois l'objet détecté, il est implicitement localisé : il est situé dans la région où la similarité a été établie. Toutefois, parce que son orientation et son échelle peuvent avoir variées ou parce qu'il peut être partiellement occulté, localiser de façon automatique ses contours exacts peut être un problème à part entière.

Dans le cas où vérifier l'existence de l'objet n'est pas suffisant et où la localisation dans l'espace est nécessaire, on doit représenter la position de l'objet sur l'acquisition. La méthode la plus simple est d'utiliser une **forme géométrique simple** (rectangles, cercles...) **pour englober** la zone concernée. L'objet est alors vulgairement localisé. Une seconde méthode est de **segmenter précisément l'objet du reste de l'image** ([88], [36], [57], [62], [37], [29]), permettant d'éviter d'inclure des éléments parasites dans la localisation (fond de la scène).

Le problème de reconnaissance peut aussi être traité de façon inverse et nous pouvons partitionner l'image pour favoriser la robustesse de la reconnaissance [83] [72]. Dans ce cas, nous pouvons **superviser en partie la segmentation** (on parle alors de segmentation semi-automatique) et utiliser des méthodes de **propagation de labels** [114, 45, 119, 8, 24]. Enfin, se baser sur un gradient, un Laplacien ou une binarisation seuillée de l'image peut aussi mettre en évidence les formes les plus simples. On pourra enfin résoudre les problèmes de mauvaises segmentations ayant échouées au niveau de détails fins ou de cas particuliers (e.g la main isolée du bras dû à la présence d'une montre) avec des opérateurs morphologiques, en reconstruisant les segmentations obtenues par dilatation (i.e **ouverture morphologique**, causant une perte de détails fin, parasites) ou par érosion (i.e **fermeture morphologique**, rendant les contours moins fidèles). Une approche particulièrement intéressante pour la segmentation et qui fonctionne de pair avec l'étape de reconnaissance, consiste à utiliser deux réseaux de neurones dédiés la segmentation et la détection simultanés d'objets [38].

2 Performance

2.1 Critères de performance d'une reconnaissance

La reconnaissance d'objets en vision par ordinateur n'est jamais parfaite et relève de l'estimation, d'où la possible nécessitée d'une assistance humaine pour limiter les erreurs (faux positifs, faux négatifs) grâce à une validation manuelle de la reconnaissance. La fiabilité du système sera évaluée au niveau des critères suivants :

— Précision,
— Rappel,
— temps d'exécution.

La **robustesse** d'un système de reconnaissance est décrite par sa capacité à résister :

— au changement d'échelle,
— à l'occultation (partielle d'un objet),
— à l'encombrement de la scène,
— à la qualité d'acquisition (bruit),
— la rotation,
— la translation.

Le **flou** (de mouvement ou de focus) constitue un problème particulier de la reconnaissance visuelle d'objets. Celui-ci fait varier drastiquement les zones d'intérêts, la texture... Par conséquent, on n'attendra pas du système qu'il soit robuste à ce genre de lissage des caractéristiques de l'image. On considérera alors que l'échec de reconnaissance est non pas dû à l'algorithme mais à l'acquisition.

On prendra en compte que la **performance humaine** est, elle aussi, imparfaite. La reconnaissance d'objets chez l'homme est correcte pour **85 à 95%** des objets évalués (sur les classifications utilisées lors des conventions d'imageries, par exemple [52]). Des expériences sur la perception humaine de similarités inter-images [90] indiquent que la **sémantique** (favorisée par la couleur) épaule notre système de reconnaissance. L'extraction de la sémantique utilise les mêmes procédés que ceux de la reconnaissance d'objets (texture, couleur, features ...) [12] et s'approche de la problématique de l'indexation d'informations visuelles [17]. Il est possible de hiérarchiser la sémantique [70] (ou les catégories [71]) des classifications

attendues dans une image afin de reconnaître les types d'objets détectés dans une image matricielle. Pour finir, la nature même de l'information à traiter influe sur la capacité de vérification du système ; il existe une différence de performance entre le traitement automatique d'image fixes et d'images séquentielles [51] [108].

2.2 Estimations & résultats

2.2.1 Système de matching par point d'intérêts

On peut s'attendre à une précision allant de **50 à 70%** avec ce type de méthode (partie 1.4.1). L'absence d'objets parasites dans le décors ainsi que la connaissance des items attendus dans notre scène favorisera grandement la reconnaissance, plaçant le problème de l'occultation en tant qu'acteur principal de la non-reconnaissance.

Chaque article propose son comparatif avec des méthodes connues (généralement SIFT et SURF), par exemple, un algorithme BRISK amènerais à un grand avantage en termes de temps de traitement pour une précision similaire lorsqu'il est lancé sur un cœur d'un quad-core i7 cadencé à 2.67Ghz sous Ubuntu 10.04 (32b) :

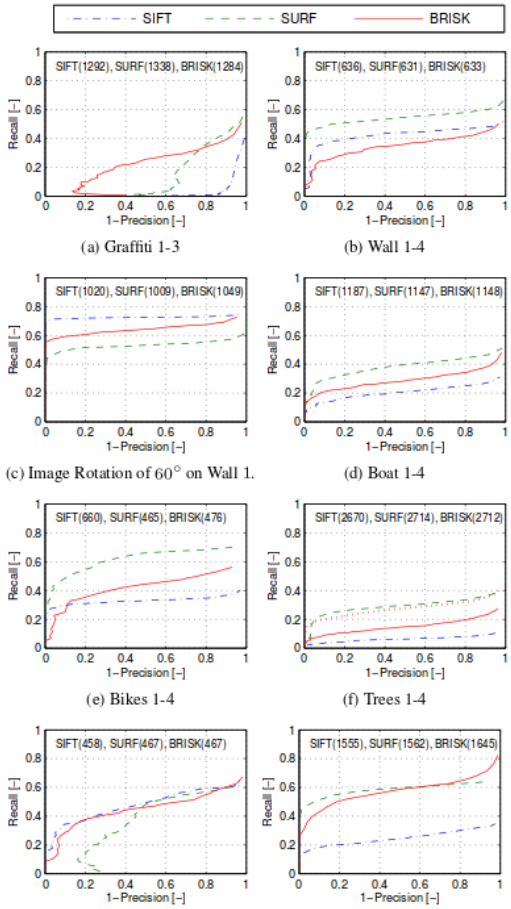
	SIFT	SURF	BRISK
Detection threshold	4.4	45700	67
Number of points	1851	1557	1051
Detection time [ms]	1611	107.9	17.20
Description time [ms]	9784	559.1	22.08
Total time [ms]	11395	667.0	39.28
Time per point (ms)	**6.156**	**0.4284**	**0.03737**

Table 1. Detection and extraction timings for the first image in the Graffiti sequence (size: 800 × 640 pixels).

	SIFT	SURF	BRISK
Points in first image	1851	1557	1051
Points in second image	2347	1888	1385
Total time [ms]	291.6	194.6	29.92
Time per comparison [ns]	**67.12**	**66.20**	**20.55**

Table 2. Matching timings for the Graffiti image 1 and 3 setup.

FIGURE 8 – Extrait 1 de l'article Brisk (2.67GHz, 32b)



(a) Graffiti 1-3 (b) Wall 1-4 (c) Image Rotation of 60° on Wall 1. (d) Boat 1-4 (e) Bikes 1-4 (f) Trees 1-4

FIGURE 9 – Extrait 2 de l'article Brisk (2.67GHz, 32b)

Des publications [76] [44] comparent efficacement une partie des détecteurs, des descripteurs et des couples {détecteurs,descripteurs} vu précédemment au niveau du temps d'exécution qu'ils requièrent :
D'après [76], la détection de points d'intérêts sur une même machine est la plus rapide pour les

détecteurs suivants :

— FAST 2ms
— ORB 7ms
— BRISK 10ms

La description des points d'intérêts, elle, préconise les descripteurs suivants en termes de temps de traitement :

— BRIEF 3.8ms
— ORB 4.2ms
— BRISK 10.6ms

Les méthodes binaires font donc parties des meilleures possibilités pour un système de reconnaissance en temps réel et leurs pourcentages de rappels et précisions sont compétitifs :

Table 4. Precision/Recall for the different descriptors and $N = 40$, $\epsilon = 3$

Detector	Descriptor	Precision	Recall	MAP
SURF	SURF	0.485	0.513	0.334
SURF	SIFT	0.525	0.533	0.491
SURF	BRIEF	0.517	0.546	0.514
SURF	ORB	0.448	0.470	0.437
SURF	LIOP	0.581	0.597	0.568
SURF	MROGH	0.540	0.567	0.527
SURF	MRRID	0.550	0.569	0.510
SURF	BRISK	0.536	0.553	0.530
BRISK	BRISK	0.504	0.527	0.492
ORB	ORB	0.493	0.495	0.463
FAST	SIFT	0.366	0.376	0.336

FIGURE 10 – Extrait de [76] (24x3.47GHz/12mb, 64b)

2.2.2 SVM, AdaBoost, RandomForest

L'utilisation de méthode d'apprentissage similaires aux SVM, AdaBoost ou RandomForest pour la reconnaissance visuelle d'objets est souvent **laborieuse**. Bien que ces solutions d'apprentissage soient beaucoup **moins coûteuses** que les réseaux de neurones, elles sont aussi beaucoup moins précises dans un cadre de reconnaissance général. En effet, elles ne sont pas robustes à la rotation, au changement d'angle ou au changement d'échelle. En revanche, elles peuvent constituer **un classifieur parfait dans un contexte invariant** (classification d'un objet depuis des paramètres de références identiques à ceux de l'acquisition que l'on souhaite évaluer). On peut mettre à profit cet aspect d'attache aux données en utilisant des SVM pour la classification intra-classe. En effet, détecter des sous-ensembles de classes génériques est une bonne mise en pratique de ces méthodes de classification. Elles constituent donc un pilier intéressant à la combinaison d'algorithmes pour la reconnaissance visuelle d'objets, notamment pour une utilisation avec les descripteurs de point d'intérêts (voir partie 1.4.1) et appréhendent convenablement que la notion d'apprentissage est au cœur de l'état de l'art.

2.2.3 Réseaux de neurones

De nos jours, les meilleures précisions pour la reconnaissance visuelle sont dues à l'utilisation d'un réseaux de neurones. L'état de l'art sur le meilleur système peuvent atteindre 81.1% de confiance (partie 1.4.3) sur de larges bases d'images. On peut s'attendre à une précision encore meilleure pour les cas où les items attendus sont restreins et la scène connue, la reconnaissance est dans ces cas proche de celle de l'homme. L'évaluation est souvent réalisée à deux niveaux pour ce type de méthode :

Top-5 error : Erreur de reconnaissance où la cible ne fait pas partie des 5 resultats les plus probables déterminés par l'algorithme.

Top-1 error : Erreur de reconnaissance où la cible n'est pas le résultat trouvé comme étant le plus probable par l'algorithme.

Modèle	Top-1 error (%)	Top-5 error (%)	NB paramS (millions)
AlexNet	37.5	17	60
VGG-A	29.6	10.4	133
VGG-D	26.8	8.7	138
ResNet-34	26.7		
VGG-E	25.5	8.0	144
BN-Inception	25.2		
ResNet-50	24.0		
ResNet-101	22.4		
Resnet-152	21.43	5.71	
Inception-Resnet-v1	21.3	5.5	
Inception-v3	21.2	5.6	5
Inception-v4	20.0	5.0	
Inception-ResNet-v2	19.9	4.9	

TABLE 1 – Comparatif des résultats des propositions entraînées (large base d'image)

2.3 ILSVRC

2.3.1 Un challenge au cœur de l'état de l'art

L'état de l'art peut être perçu comme le fruit de l'*Imagenet Large Scale Visual Recognition Challenge* [94]. Il s'agit d'un **concours** mettant en compétition les plus grands acteurs en reconnaissance d'objets (aussi bien des entreprises comme Google ou Microsoft que des groupes de chercheurs). Les résultats de la compétition ainsi que les méthodes/algorithmes utilisés sont pour la plupart ouverts et accessibles au public. Comme son nom l'indique, la problématique concerne de **larges bases d'images** dont le contenu est inconnu par les algorithmes en pré-traitement. Le classement des résultats est alors réalisée selon plusieurs critères comme par exemple l'importance de la base d'entraînement mise à disposition. Certaines propositions ne sont pas pertinentes pour de la reconnaissance d'objets connus dans un cadre dont ont connaît les paramètres (luminosité, échelle) mais s'adresse plutôt à de la reconnaissance d'objets quelconques dans un cadre inconnu. Il est intéressant d'observer l'évolution de la nature des entrées proposées par les participants. En effet, on peut alors constater que la communauté scientifique qui approchait les problèmes de reconnaissance par diverses méthodes (Perceptron, SVM, descripteur de points d'intérêts, mots visuels entre autres) a, au cour des dernières décennies, été unanimement convaincue par l'apprentissage profond et utilise aujourd'hui exclusivement des variantes de réseaux de neurones pour la plupart convolutifs [33]. Plus d'informations et notamment les résultats chiffrés de l'ILSVRC 2015 sont disponibles sur le site web `http://image-net.org/challenges/LSVRC/2015/`. Dernièrement, les notions de "batch normalization" (normalisation des lots) et de neurones résiduels (capables de sauter des connexions) ont permis d'améliorer les résultats de l'état de l'art en limitant le nombre de calculs nécessaires pour converger vers la capacité de reconnaissance optimale et donc d'augmenter la vitesse d'apprentissage.

FIGURE 11 – Imagenet LSVRC

2.3.2 Propositions sur la reconnaissance d'objets

Pour la compétition qui a pris place en 2015, la meilleure équipe a réussi à obtenir une précision moyenne d'environ **60 à 62%** et possède les meilleurs taux de classification et localisation d'objets sur de larges bases d'images. L'équipe MRSA (Microsoft Research Asia) décrit alors sa méthode de la manière suivante :

We train neural networks with depth of over 150 layers. We propose a "deep residual learning" framework [a] that eases the optimization and convergence of extremely deep networks. Our "deep residual nets" enjoy accuracy gains when the networks are substantially deeper than those used previously. Such accuracy gains are not witnessed for many common networks when going deeper.

Our localization and detection systems are based on deep residual nets and the "Faster R-CNN" system in our NIPS paper [b]. The extremely deep representations generalize well, and greatly improve the results of the Faster R-CNN system. Furthermore, we show that the region proposal network (RPN) in [b] is a generic framework and performs excellent for localization.

We only use the ImageNet main competition data. We do not use the Scene/VID data.

The details will be disclosed in a later technical report of [a].

[a] "Deep Residual Learning for Image Recognition", Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Tech Report 2015. [b] "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks", Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. NIPS 2015.

2.3.3 Propositions sur la reconnaissance dans un flux vidéo

L'équipe **CUVideo** obtient les meilleurs résultats sur la reconnaissance d'objets dans un flux vidéo avec une précision moyenne de **67%** environ. Selon la base d'entrainement, **Amax** obtient aussi de bon résultats (précision moyenne à **73%**).

Description de la méthode de l'équipe **Amax** (reconnaissance vidéo uniquement) :

2.VID : Tempor Cascade region regression

Objectiveness based tracker is designed to track the objects on videos.

Firstly, We train Faster-RCNN[3](VGG-16) with the provided training data (sampling from frames). The network provides features for tracking.

Secondly,the tracker uses the roi_pooling features from the last conv layer and tempor information, which can be seen as the tempor Fast RCNN[2]. (Take the location-indexed features from current frame to predict the bounding box of object on next frame.)

Tempor information and scence cluster(different video from one scence) are greatly helpful to decide the classes on the videos with high confidence.

[1]Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014 : 580-587. [2]Girshick R. Fast R-CNN[J]. arXiv preprint arXiv :1504.08083, 2015. [3]Ren S, He K, Girshick R, et al. Faster r-cnn : Towards real-time object detection with region proposal networks[J]. arXiv preprint arXiv :1506.01497, 2015.

Description de la méthode de l'équipe **CUVidéo** :

For object detection in video, we first employ CNN based on detectors to detect and classify candidate regions on individual frames. Detectors are based on the combination of two types of models, i.e. DeepID-Net [a] in ILSVRC 2014 and Faster RCNN [b]. The temporal information is employed propagating detection scores. Score propagation is based on optical flow estimation and CNN based tracking [c]. Spatial and temporal pool along tracks is employed. Video context is also used to rescore candidate regions.

[a] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C. Loy, X. Tang, "DeepID-Net : Deformable Deep Convolutional Neural Networks for Object Detection," CVPR 2015. [b] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," arXiv :1506.01497. [c] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual Tracking with Fully Convolutional Networks," ICCV 2015.

2.4 LPIRC

2.4.1 La reconnaissance visuelle au moindre coût

Récemment, les recherches en matières de reconnaissances visuelles étaient orientées sur la précision générale des solutions. Ainsi, celles-ci ne s'arrêtaient pas à la nécessité de fonctionner sur du matériel ultra-performant, dédié et peu abordable. Cependant, avec l'amélioration des résultats, les publications démontrent maintenant une volonté progressive de limiter ces prérequis, en limitant le nombre de paramètres dans les réseaux de neurones ou en simplifiant l'apprentissage. Ainsi, le *Low-Power Image Recognition Challenge* évalue les entrées du concours selon deux critères : une grande précision pour une faible consommation d'énergie. On cible alors les unités de calculs de plus faible capacités et limitées par une batterie comme par exemple, celles présentes sur les smartphones. Plus d'informations sur les limitations du concours sur `http://www.lpirc.net`.

FIGURE 12 – IEEE LPIRC

2.5 CVPR

La conférence sur la *"**C**omputer **V**ision and **P**attern **R**ecognition"* (CVPR) est une conférence scientifique annuelle de l'IEEE en reconnaissance des formes et vision par ordinateur. Certains articles dont nous avons parlés dans cet état de l'art y ont été présentés. De nombreuses propositions sont envoyés (cette année 1865 ont été parcourus et validés) et très peux y sont présentés (83 présentation orales, 123 présentations éclairs sur les points saillants de l'article et le reste sous forme de posters interactifs). La compétition a eu lieu à Las Vegas (États-Unis) du 24 Juin au 2 Juillet et a permis de mettre en avant les points suivants :

FIGURE 13 – IEEE CVPR

3 Utilisation d'un système de reconnaissance

3.1 Supervision

Bien que les solutions proposées possèdent une bonne précision, la présence d'un utilisateur du système peut être justifiée et il faudra tenir compte que certains aspects méritent son attention pour garantir une bonne précision lors de la reconnaissance :

- **Orientation** : La présence de boutons, d'étiquettes sur tout type d'objets facilite la reconnaissance. On peut alors demander aux utilisateurs de retourner ou non certains objets (opération très peu coûteuse en temps).
- **Séparation** : Dans la mesure où une segmentation manuelle des objets facilite la reconnaissance, le replacement des items à identifier dans le champ d'acquisition peut être demandé.
- **Occultation et bruit** : On demandera à l'utilisateur d'éviter autant que possible tout type d'occultation dans le champ de vision de la caméra pendant la reconnaissance des items. On demandera aussi d'écarter du champ d'acquisition tout type d'objets susceptible de générer des faux-positifs lors de la reconnaissance (stylo, boite en carton etc.).
- **Initialisation** : Selon l'algorithme choisi, il est possible que l'utilisateur ait à initialiser le système en début de journée. Cela consisterait à capturer la scène sans aucun objet (prise d'une photo par la caméra) et permettrais au système de soustraire à celle-ci d'éventuelles rayure/dégât affligés au poste par l'usure qui pourraient entraver la reconnaissance. Le temps nécessaire à une telle procédure serait presque négligeable.

3.2 Vulnérabilités

Sensible à l'environnement qu'il acquiert, nous verrons dans ce dernier point qu'un système de reconnaissance connaît des vulnérabilités qui s'étendent au-delà de l'aspect technique de la solution.

3.2.1 Systèmes d'apprentissage

Un système de reconnaissance à l'**apprentissage finit** se trompera régulièrement (bruit, complexité, autre...) pour des causes environnementales. Ces perturbations peuvent être issues du hasard (composition d'un effet d'optique), d'un souci technique (vieillissement du matériel, choc), ou d'un facteur étranger (insecte sur l'objectif, humidité, rayon de lumière "aveuglant", etc.). Ces phénomènes étant reproductibles, on peut **gêner** volontairement une reconnaissance. Plus problématique, il est possible de la **tromper**. La vulnérabilité la plus évidente et simple à mettre en place consiste à profiter des caractéristiques de la vision du système pour la forcer à reconnaître un élément qui n'existe pas. On peut, par exemple, créer "replay" depuis une impression papier de l'objet concerné []. Il est aussi possible de manipuler l'environnement et/ou l'image traitée pour complètement empêcher une reconnaissance utilisant un système d'apprentissage là où un humain n'aurai connu aucune difficulté à en identifier les éléments (sur l'original autant que sur la modification) [53]. Ce genre de manipulation remet en question l'utilisation de système de reconnaissance par apprentissage à des fins de sécurité (caméra surveillance reconnaissant des items dangereux) ou de vérification d'intégrité (surveillance d'objets stockés).

Un système de reconnaissance à l'**apprentissage continu** se trompera, dans les conditions idéales, de moins en moins. Ces systèmes apprennent et s'adaptent dans le temps en fonction de l'évolution d'une cible (comportement ou choix d'un utilisateur, par exemple). La vulnérabilité de ce système est qu'il est exposé à la manipulation de la connaissance qui pourra être généralisée à des cas trop larges. Prenons, à titre d'exemple, une reconnaissance basée sur un système d'apprentissage continu cherchant à identifier les comportements dangereux de personnes dans une rue. La venue temporaire d'un groupe d'artistes effectuant une représentation, de vendeurs lors d'une brocante, de moyens de locomotions variants, de personnes courants, jouant, tombants, transportant des objets de plusieurs façons ... viendra ajouter un grand nombre de cas estimés comme non dangereux. On peut alors se poser la question de savoir si l'apprentissage continu passera ou non à côté du comportement dangereux. Enfin, un individu averti pourra utiliser cette vulnérabilité en mimant pour camoufler une action dangereuse dans une autre afin de la rendre anodine ; dès lors, le système de reconnaissance échoue.

Les vulnérabilités de cette famille de systèmes de reconnaissance face à un adversaire ("adversarial machine learning") n'ont été que très récemment explorées, mais peuvent d'ors et déjà classées en différents types similaires à ceux établis dans les problématiques de régulation de spam, par exemple. La

problématique d'un point de vue sécurité est traitée par B.Biggio [9].

4 Documents liés au sujet non cités

1. Rich feature hierarchies for accurate object detection and semantic segmentation [34]
2. Learning a Deep Compact Image Representation for Visual Tracking [35]
3. Listing d'applications d'algorithme de vision par ordinateur selon David Lowe, créateur de SIFT [66]
4. Selection of Discriminative Regions and Local Descriptors for Generic Object Class Recognition [25]
5. Synthetic 3D Model-Based Object Class Detection and Pose Estimation [65]
6. A semi-supervised learning approach to object recognition with spatial integration of local features and segmentation cues [16]
7. Fisher kernel based models for image classification and object localization [20].
8. Local features and kernels for classification of texture and object categories : a comprehensive study [118]
9. Contextual Object Detection using Set-based Classification [21]
10. Learning object class detectors from weakly annotated video [86]
11. Object class recognition by unsupervised scale-invariant learning [31]

Références

[1] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG)*, 34(6) :219, 2015.

[2] Fadel Adib and Dina Katabi. *See through walls with wifi!*, volume 43. ACM, 2013.

[3] Abdalrahman Al-qubaa. An electromagnetic imaging system for metallic object detection and classification. 2013.

[4] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast Retina Keypoint. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] Pablo F. Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE Features. *Université d'Auvergne, Clermont Ferrand, France. Department of Computing, Imperial College London, UK.*

[6] Luís A Alexandre. 3d object recognition using convolutional neural networks with transfer learning between input channels. In *Intelligent Autonomous Systems 13*, pages 889–898. Springer, 2016.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). septembre 2008.

[8] Serge Beucher and Fernand Meyer. The morphological approach to segmentation : the watershed transformation. *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 34 :433–433, 1992.

[9] Battista Biggio. Machine learning under attack : Vulnerability exploitation and security measures. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 1–2. ACM, 2016.

[10] JF BONNET, D DUCLOS, R SAMY, and G STAMON. Poursuite et reconnaissance d'objets dans une séquence infrarouge par contours actifs robustes incrémentaux et réseaux neuronaux. In *16e Colloque sur le traitement du signal et des images, FRA, 1997*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 1997.

[11] Sabri Boughorbel, Jean-Philippe Tarel, and Francois Fleuret. Non-Mercer Kernels for SVM Object Recognition. In *BMVC*, pages 1–10. Citeseer, 2004.

[12] Ben Bradshaw. Semantic based image retrieval: a probabilistic approach. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 167–176. ACM, 2000.

[13] Gilles Burel. *RESEAUX DE NEURONES EN TRAITEMENT D'IMAGES - Des Modèles theoriques aux Applications Industrielles*. PhD thesis, Septembre 1991.

[14] Pierre Buyssens, Marinette Revenu, and Olivier Lepetit. Réseau de Neurones Convolutionnels pour la Reconnaissance Faciale Infrarouge. In *XXIIe colloque GRETSI (traitement*

du signal et des images), Dijon (FRA), 8-11 septembre 2009. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2009.

[15] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. *Computer Vision–ECCV*, 2010.

[16] Peter Carbonetto, Gyuri Dorkó, Cordelia Schmid, Hendrik Kück, and Nando De Freitas. A semi-supervised learning approach to object recognition with spatial integration of local features and segmentation cues. In Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors, *Towards category-level object recognition*, volume 4170 of *Lecture Notes in Computer Science (LNCS)*, pages 277–300. Springer, 2006. URL : `https://hal.inria.fr/inria-00548613`, `doi:10.1007/11957959\_15`.

[17] Y Chahir. *Indexation et recherche par le contenu d'informations visuelles*. PhD thesis, Thèse de Doctorat, ICTT, Lyon, France, 2000.

[18] Meixi Chen, Yule Yuan, and Yong Zhao. KAZE FeaturePoint with Modified-SIFT Descriptor. *3rd International Conference on Multimedia Technology*, 2013.

[19] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-Sweep: Extracting Editable Objects from a Single Photo. *Tsinghua University*.

[20] Ramazan Gokberk Cinbis. *Fisher kernel based models for image classification and object localization*. Theses, Université de Grenoble, July 2014. URL : `https://tel.archives-ouvertes.fr/tel-01071581`.

[21] Ramazan Gokberk Cinbis and Stan Sclaroff. Contextual Object Detection using Set-based Classification. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV 2012 - European Conference on Computer Vision*, volume 7577, pages 43–57, Firenze, Italy, October 2012. Springer. URL : `https://hal.inria.fr/hal-00756638`, `doi:10.1007/978-3-642-33783-3\_4`.

[22] P. Daum, J.L. Buessler, and J.P. Urban. Reconnaissance d'objets par apprentissage d'images – Réseaux de neurones à champs récepteurs aléatoires. *RFIA (Reconnaissance des Formes et Intelligence Artificielle)*.

[23] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez. Revisiting the VLAD image representation. In *ACM Multimedia*, Barcelona, Spain, October 2013. URL : `https://hal.inria.fr/hal-00840653`.

[24] Lei Ding and Alper Yilmaz. Enhancing interactive image segmentation with automatic label set augmentation. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010.

[25] Gyuri Dorkó. *Selection of Discriminative Regions and Local Descriptors for Generic Object Class Recognition*. Theses, Institut National Polytechnique de Grenoble - INPG, June 2006. URL : `https://tel.archives-ouvertes.fr/tel-00555064`.

[26] Gyuri Dorko and Cordelia Schmid. object class recognition using discriminative local features. 2005.

[27] Jean-Denis DUROU, Xavier DESCOMBES, Pavel LUKASHEVISH, and Aliaxandr KRAUSHONAK. Reconstruction 3D du bâti à partir d'une seule image par naissances et morts multiples. *ACM Transactions on Graphics (TOG)*, 32(6) :195, 2013.

[28] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[29] Mohammed El Hassani, Stéphanie Jehan-Besson, Luc Brun, Marinette Revenu, Marc Duranton, David Tschumperlé, and Delphine Rivasseau. A time-consistent video segmentation algorithm designed for real-time implementation. *VLSI Design*, 2008(2) :7, 2008.

[30] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable Object Detection using Deep Neural Networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[31] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

[32] Vittorio Ferrari, Frédéric Jurie, and Cordelia Schmid. Accurate Object Detection with Deformable Shape Models Learnt from Images. In *CVPR 2007 - Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, United States, June 2007. IEEE Computer society. URL : `https://hal.archives-ouvertes.fr/hal-00203920`, `doi:10.1109/CVPR.2007.383043`.

[33] Efstratios Gavves. A brief history of Computer Vision (and not time). http://www.egavves.com/, Decembre 2014.

[34] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *UC Berkeley*.

[35] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Learning a Deep Compact Image Representation for Visual Tracking. pages 580–587, 2014.

[36] Daniele D Giusto, Francesco Massidda, and Cristian Perra. FACE: fast active-contour curvature-based evolution. *Signal Processing : Image Communication*, 19(6) :517–538, 2004.

[37] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110(3) :328–348, 2014.

[38] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[39] Chris Harris and Mike Stephens. A COMBINED CORNER AND EDGE DETECTOR. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL : http://arxiv.org/abs/1512.03385.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[42] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3) :425–436, 2009.

[43] Scott Helmer and David Lowe. Using stereo for object recognition. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3121–3127. IEEE, 2010.

[44] Antti Hietanen, Jukka Lankinen, Joni-Kristian Kämäräinen, Anders Glent Buch, and Norbert Krüger. A comparison of feature detectors and descriptors for object class matching. *Neurocomputing*, 2015.

[45] Nawal Houhou, Xavier Bresson, Arthur Szlam, Tony F Chan, and Jean-Philippe Thiran. Semi-supervised segmentation based on non-local continuous min-cut. In *Scale Space and Variational Methods in Computer Vision*, pages 112–123. Springer, 2009.

[46] Andrew G. Howard. Some improvements on Deep Convolutional Neural Network Based Image Classification. *Andrew Howard Consulting*.

[47] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*, 2015.

[48] Tingting Jiang, Frédéric Jurie, and Cordelia Schmid. Learning Shape Prior Models for Object Matching. In *CVPR 2009 - IEEE Conference on Computer Vision & Pattern Recognition*, pages 848–855, Miami, United States, June 2009. IEEE Computer Society. URL : https://hal.inria.fr/inria-00548646, doi:10.1109/CVPR.2009.5206568.

[49] Andrew E. Johnson and Martial Hebert. Recognizing Objects by Matching Oriented Point. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[50] Frédéric Jurie and Cordelia Schmid. Scale-invariant shape features for recognition of object categories. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–90. IEEE, 2004.

[51] Vicky Kalogeiton, Vittorio Ferrari, and Cordelia Schmid. Analysing domain shift factors between videos and images for object detection. working paper or preprint, January 2016. URL : https://hal.inria.fr/hal-01281069.

[52] Andrej Karpathy. What I learned from competing against a ConvNet on ImageNet. Blog, GITHUB, Septembre 2014. URL : http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-

[53] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv :1607.02533*, 2016.

[54] Akash Kushal, Cordelia Schmid, and Jean Ponce. Flexible object models for category-level 3d object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[55] Diane Larlus, Gyuri Dorkó, and Frédéric Jurie. Création de vocabulaires visuels efficaces

pour la catégorisation d'images. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA'06)*, 2006.

[56] François Lecellier, Stéphanie Jehan-Besson, and J Fadili. Statistical region-based active contours for segmentation : an overview. *IRBM*, 35(1) :3–10, 2014.

[57] François Lecellier, Stéphanie Jehan-Besson, and Jalal M. Fadili. Statistical region-based active contours for segmentation: an overview. *Innovation and research in biomedical engineering*, 35(1) :3–10, 2014. URL : https://hal.archives-ouvertes.fr/hal-00918290.

[58] A. Lechervy, P h. Gosselin, and F. Precioso. Active boosting for interactive object retrieval. In *International Conference on Pattern Recognition (ICPR)*, page 1, Istanbul, Turkey, Aug. 2010. URL : http://hal.archives-ouvertes.fr/hal-00520294.

[59] A. Lechervy, P h. Gosselin, and F. Precioso. Boosting kernel combination for multi-class image categorization. In *IEEE International Conference on Image Processing (ICIP)*, page 1, Orlando, Florida, U.S.A, September 2012. URL : http://hal.archives-ouvertes.fr/index.php?halsid=74137bb9dsjl8lu5475jh13h33&view_this_doc=hal-00753156&version=1.

[60] Alexis Lechervy. *Interactive and multi-class Learning to detect semantic concepts in the multimedia data.* Theses, Université de Cergy Pontoise, December 2012. URL : https://tel.archives-ouvertes.fr/tel-01087070.

[61] Alexis Lechervy, Philippe-Henri Gosselin, and Frédéric Precioso. Active Boosting for interactive object retrieval. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3268–3271. IEEE, 2010.

[62] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011.

[63] Stefan Leutenegger, Margarita Chli, and Roland Y.Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. *Computer Vision (ICCV)*, 2011.

[64] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. *arXiv preprint arXiv :1603.02199*, 2016.

[65] Joerg Liebelt. *Synthetic 3D Model-Based Object Class Detection and Pose Estimation.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2010.

[66] David Lowe. The computer vision industry. URL : http://www.cs.ubc.ca/~lowe/vision.html.

[67] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.

[68] C. MAAOUI, H. LAURENT, and B. EMILE. Reconnaissance et détection robuste d'objets couleur. *GRETSI, Groupe d'Etudes du Traitement du Signal et des Images.*

[69] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of Exemplar-SVMs for Object Detection and Beyond. *IEEE International Conference on Computer Vision (ICCV)*, pages 89–96, Novembre 2011.

[70] Marcin Marszalek and Cordelia Schmid. Semantic Hierarchies for Visual Object Recognition. In *CVPR 2007 - IEEE Conference on Computer Vision & Pattern Recognition*, pages 1–7, Minneapolis, United States, June 2007. IEEE Computer Society. URL : https://hal.inria.fr/inria-00548680, doi:10.1109/CVPR.2007.383272.

[71] Marcin Marszalek and Cordelia Schmid. Constructing Category Hierarchies for Visual Recognition. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *ECCV 2008 - 10th European Conference on Computer Vision*, volume 5305 of *Lecture Notes in Computer Science*, pages 479–491, Marseille, France, October 2008. Springer-Verlag. URL : https://hal.inria.fr/inria-00548656, doi:10.1007/978-3-540-88693-8_35.

[72] Marcin Marszalek and Cordelia Schmid. Accurate Object Recognition with Shape Masks. *International Journal of Computer Vision*, 97(2) :191–209, April 2012. URL : https://hal.inria.fr/hal-00650941, doi:10.1007/s11263-011-0479-2.

[73] D. Maturana and S. Scherer. VoxNet : A 3D Convolutional Neural Network for Real-Time Object Recognition. In *IROS*, 2015.

[74] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10) :1615–1630, 2005.

[75] Krystian Mikolajczyk, Andrew Zisserman, and Cordelia Schmid. Shape recognition with edge-based features. In Richard Harvey and Andrew Bangham, editors, *British Machine Vision Conference (BMVC '03)*, volume 2,

pages 779–788, Norwich, United Kingdom, September 2003. The British Machine Vision Association. URL : https://hal.inria.fr/inria-00548226.

[76] Ondrej Miksik and Krystian Mikolajczyk. Evaluation of Local Detectors and Descriptors for Fast Feature Matching. *21st International Conference on Pattern Recognition (ICPR)*, pages 2681 – 2684, Novembre 2012.

[77] Hirosi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14 :5–24, 1995.

[78] URATEK MUSEUM. Mémorisation d'attracteur par réseau de neurone. old URATEK website. URL : http://uratek.free.fr/research/theorie/theorie.shtml?Contr%F4le%20de%20qualit%E9%20%28UraDyco%29.

[79] Gaurav K Nanani and MVV Kantipudi. A Study of WI-FI based System for Moving Object Detection through the Wall. *International Journal of Computer Applications*, 79(7), 2013.

[80] Eric Nowak. *Reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentation locales*. PhD thesis, Mars 2008.

[81] Indian Institute of Technology Madras. Local Feature Detectors and Descriptors. *CS 6350 - COMPUTER VISION*.

[82] Pablo F. Alcantarilla and Jesús Nuevo and Adrien Bartoli. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. *School of Interactive Computing, Georgia Institute of Technology, TrueVision Solutions, Université d'Auvergne*, 2011.

[83] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert. Object Recognition by Integrating Multiple Image Segmentations. In David A. Forsyth and Philip H. S. Torrand Andrew Zisserman, editors, *ECCV 2008 - 10th European Conference on Computer Vision*, volume 5304 of *Lecture Notes in Computer Science*, pages 481–494, Marseille, France, October 2008. Springer-Verlag. URL : https://hal.inria.fr/inria-00548655, doi:10.1007/978-3-540-88690-7_36.

[84] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages 82–90, 2014.

[85] Jean Ponce, Svetlana Lazebnik, Fredrick Rothganger, and Cordelia Schmid. Toward true 3D object recognition. In *Reconnaissance de Formes et Intelligence Artificielle*, 2004.

[86] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. learning object class detectors from weakly annotated video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3282–3289. IEEE, 2012.

[87] Charles R Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. *arXiv preprint arXiv :1604.03265*, 2016.

[88] Xiaofeng Ren and Chunhui Gu. Figure-Ground Segmentation Improves Handled Object Recognition in Egocentric Video. *Computer Vision and Pattern Recognition*, Juin 2010.

[89] Jérôme Revaud, Yasuo Ariki, Guillaume lavoué, and Atilla baskurt. Combinaison de caractéristiques pour la reconnaissance rapide, robuste et invariante d'objets spécifiques. *Université de Lyon, Université de Kobe*.

[90] Bernice E Rogowitz, Thomas Frese, John R Smith, Charles A Bouman, and Edward B Kalin. Perceptual image similarity experiments. In *Photonics West'98 Electronic Imaging*, pages 576–590. International Society for Optics and Photonics, 1998.

[91] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints. *International Journal of Computer Vision*, 66(3) :231–259, 2006. URL : https://hal.inria.fr/inria-00548618, doi:10.1007/s11263-005-3674-1.

[92] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB : an efficient alternative to SIFT or SURF. *IEEE International Conference on Computer Vision (ICCV)*, Novembre 2011.

[93] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115 :211–252, Décembre 2015.

[94] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3) :211–252, 2015.

[95] Payam Saisan, Gianfranco Doretto, Ying Nian Wu, and Stefano Soatto. Dynamic

texture Recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[96] Nizar Sallem and Michel Devy. Modélisation d'Objets 3D en vue de leur reconnaissance et leur manipulation par un robot personnel Modelling of 3D objects for their recognition and manipulation by a companion robot. In *ORASIS'09-Congrès des jeunes chercheurs en vision par ordinateur*, 2009.

[97] Cordelia Schmid and Roger Mohr. Combining Greyvalue Invariants with Local Constraints for Object Recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pages 872–877, San Francisco, United States, June 1996. IEEE Computer society. URL : `https://hal.inria.fr/inria-00548368`, `doi:10.1109/CVPR.1996.517174`.

[98] Evan Shelhamer, Jonathon Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. 2016.

[99] B Smith and R Gosine. Support vector machines for object recognition, 2001.

[100] Marcelo Borghetti Soares, Pablo Barros, German I Parisi, and Stefan Wermter. Learning objects from rgb-d sensors using point cloud-based neural networks. In *Proceedings*, page 439. Presses universitaires de Louvain, 2015.

[101] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. In *Advances in Neural Information Processing Systems*, pages 665–673, 2012.

[102] Shuran Song and Jianxiong Xiao. Sliding shapes for 3D object detection in depth images. In *Computer Vision–ECCV 2014*, pages 634–651. Springer, 2014.

[103] Bruno STUNER, Clément CHATELAIN, and Thierry PAQUET. Cascade de réseaux blstm vérifiée par le lexique pour la reconnaissance d'écriture.

[104] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015.

[105] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. URL : `http://arxiv.org/abs/1602.07261`.

[106] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep Neural Networks for Object Detection. *Advances in Neural Information Processing Systems 26 (NIPS)*, 2013.

[107] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. Rethinking the Inception Architecture for Computer Vision. *University College London*, September.

[108] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 638–646, 2012.

[109] Alexander Thomas, Vittorio Ferrar, Bastian Leibe, Tinne Tuytelaars, Bernt Schiel, and Luc Van Gool. https ://hal.inria.fr/inria-00548577/document Towards multi-view object class detection. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1589–1596. IEEE, 2006.

[110] Felipe De Jesus Trujillo-Romero. *Modélisation et reconnaissance active d'objets 3D de forme libre par vision en robotique*. Theses, Institut National Polytechnique de Toulouse - INPT, December 2008. URL : `https://tel.archives-ouvertes.fr/tel-00842693`.

[111] G Turhan-Sayan and E Ergin. Electromagnetic Object Recognition for Dielectric Coated Conductors Based on WD-PCA Type Fused Feature Extraction. *Session 1P3a Theory and Methods of Digital Signal and Image Processing 2*, page 121.

[112] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[113] Deepak Geetha Viswanathan. Features from Accelerated SegmentTest(FAST). 2009.

[114] Fei Wang, Xin Wang, and Tao Li. Efficient label propagation for interactive image segmentation. In *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pages 136–141. IEEE, 2007.

[115] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local Intensity Order Pattern for Feature Description. *IEEE International Conference on Computer Vision (ICCV)*, pages 603–610, Novembre 2011.

[116] Oksana Yakhnenko, Jakob Verbeek, and Cordelia Schmid. Region-Based Image Classification with a Latent SVM Model. Research Report RR-7665, INRIA, July 2011. URL : `https://hal.inria.fr/inria-00605344`.

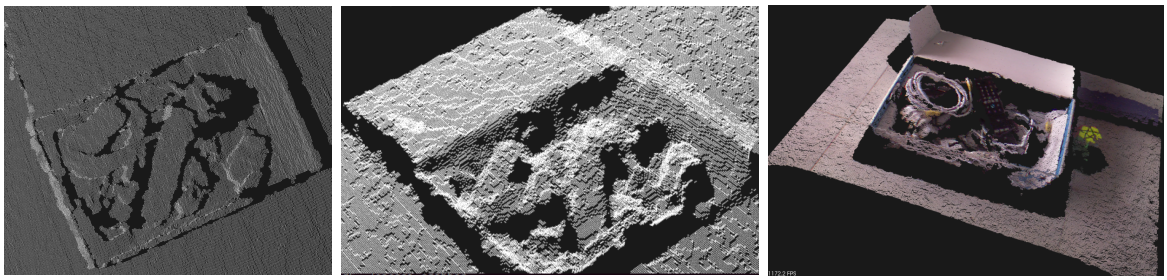
[117] Xin Yang and Kwang-Ting Cheng. LDB: An Ultra-Fast Feature for Scalable Augmen-

ted Reality on Mobile Devices. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 49–57, Novembre 2012.

[118] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2) :213–238, June 2007. URL : `https://hal.archives-ouvertes.fr/hal-00171412`, `doi:10.1007/s11263-006-9794-4`.

[119] Lei Zhang, Xun Wang, Nicholas Penwarden, and Qiang Ji. An image segmentation framework based on patch segmentation fusion. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 187–190. Ieee, 2006.

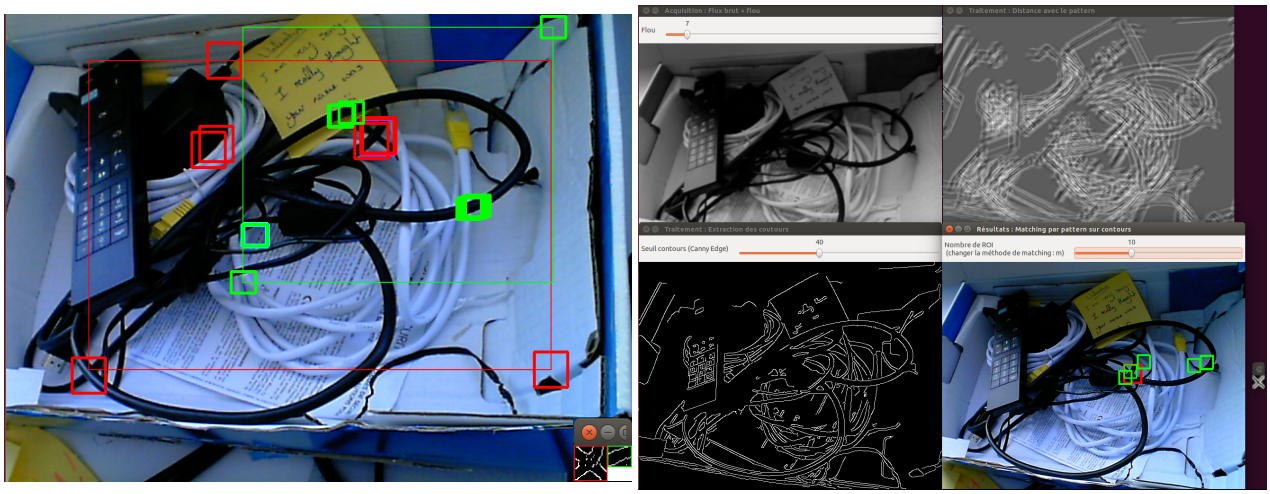
Annexe 2 : Corps de scènes



(a) vue de dessus (b) vue de biais (c) vue d'ensemble

FIGURE 42 – Représentation en nuages de points d'une acquisition, utilisant PCL, OpenNi et Kinect pour acquérir notre scène en profondeur.

Annexe 3 : Résultats d'approches structurelles



(a) Exemple de résultat

(b) Exemple de mauvaise détection

FIGURE 43 – Approche structurelle, matching

FIGURE 44 – Gradient de notre scène en utilisant la méthode de Laplace

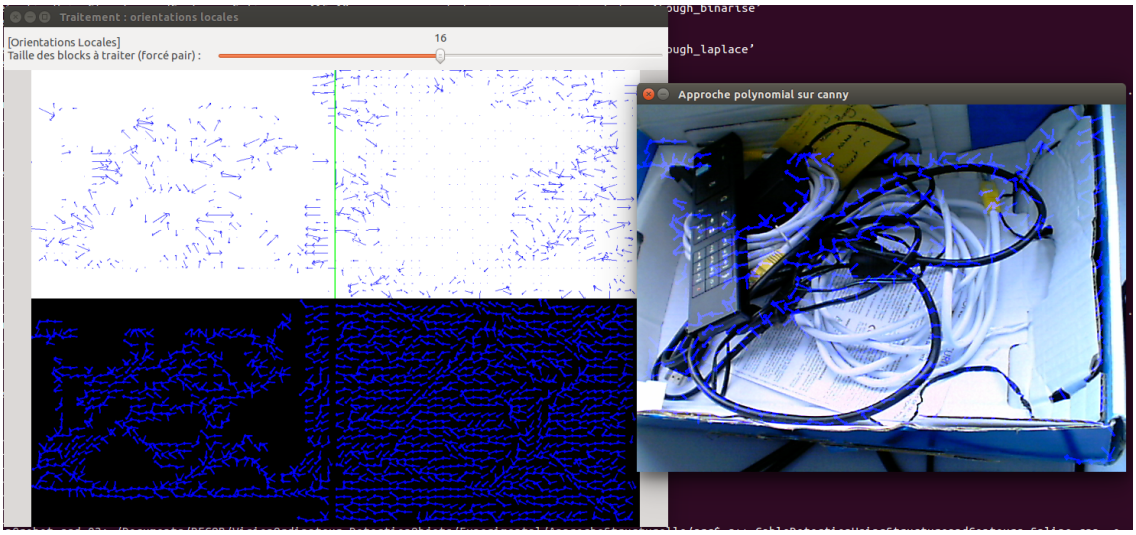


FIGURE 45 – Différentes approches d'orientations locales

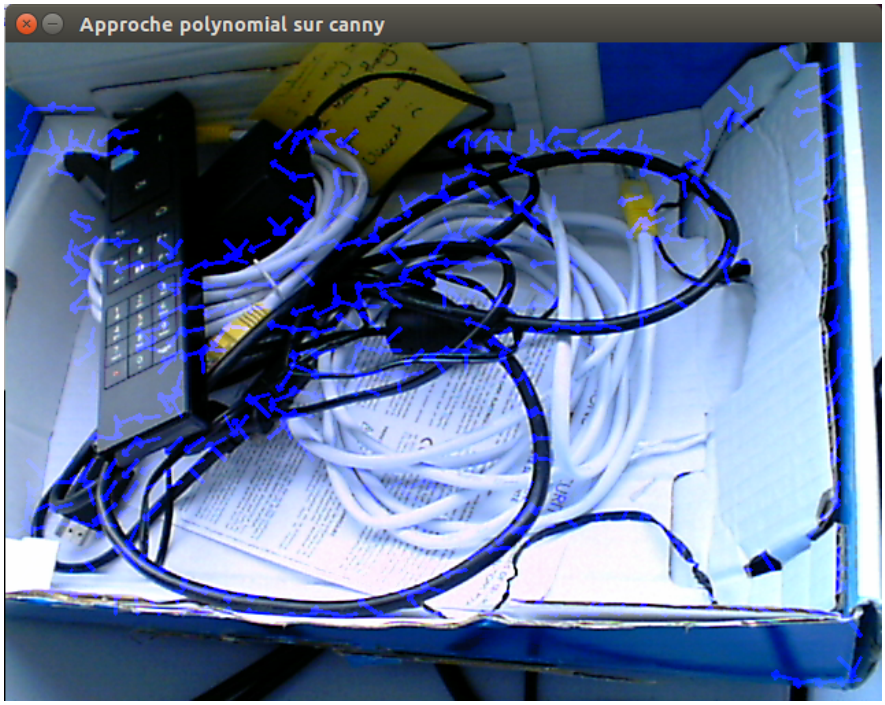
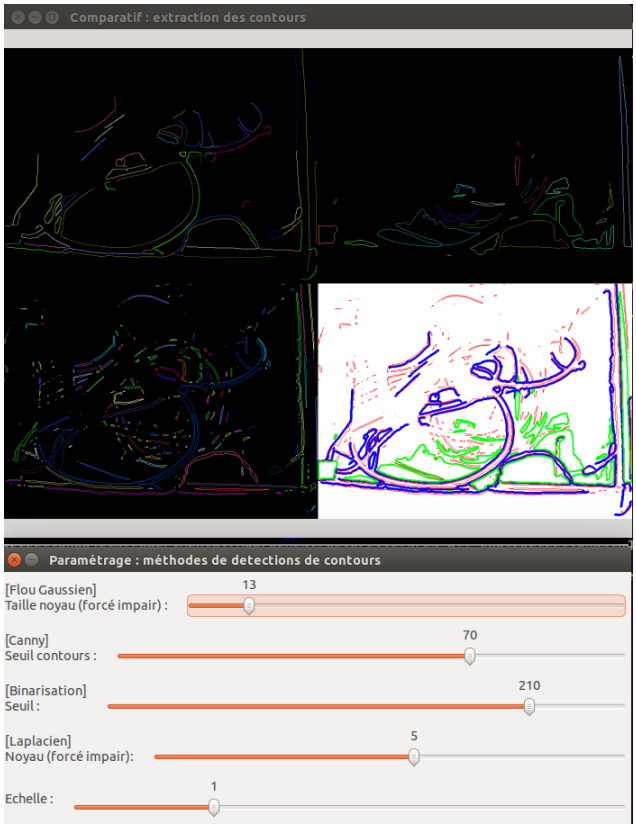
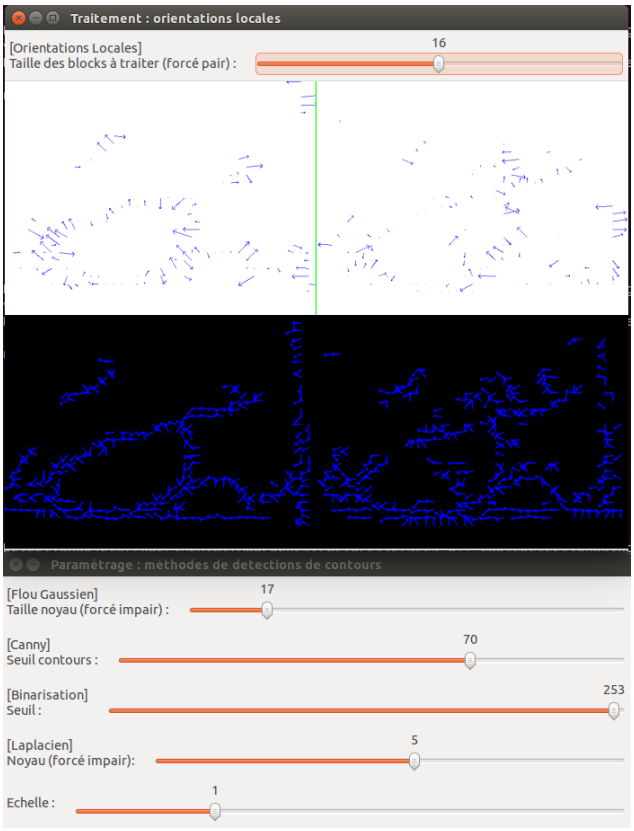


FIGURE 46 – Approche polynomial sur canny



(a) Combinaison de détection de contours



(b) Exemple intermédiaire de détection d'orientations locales sur des câbles par deux approches

FIGURE 47 – Combinaison

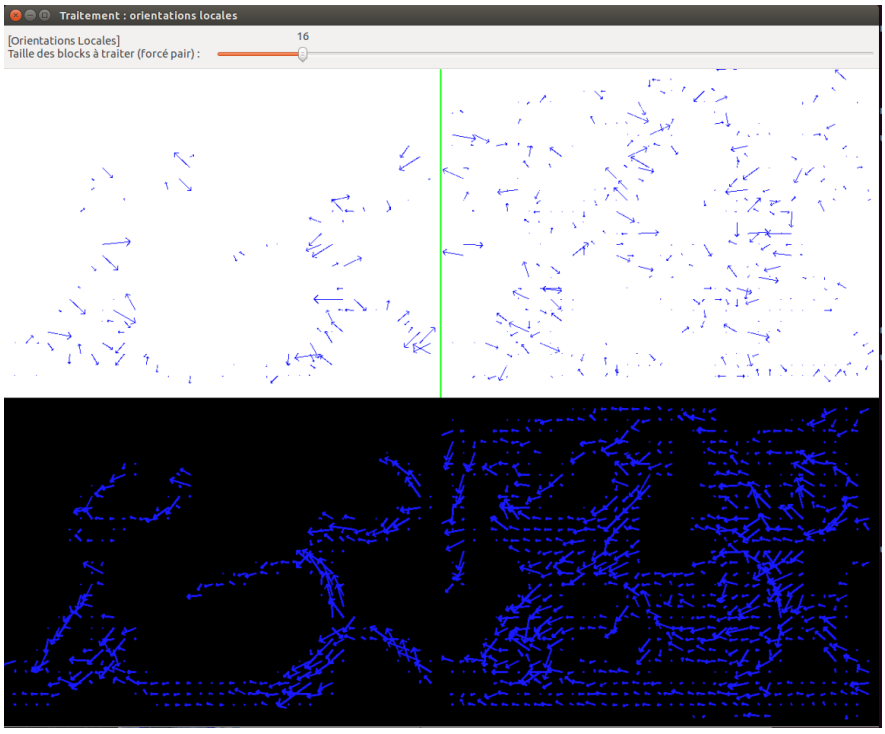


FIGURE 48 – Exemple de détection d'orientations locales avec magnitudes sur des câbles par deux approches sur deux détections de contours

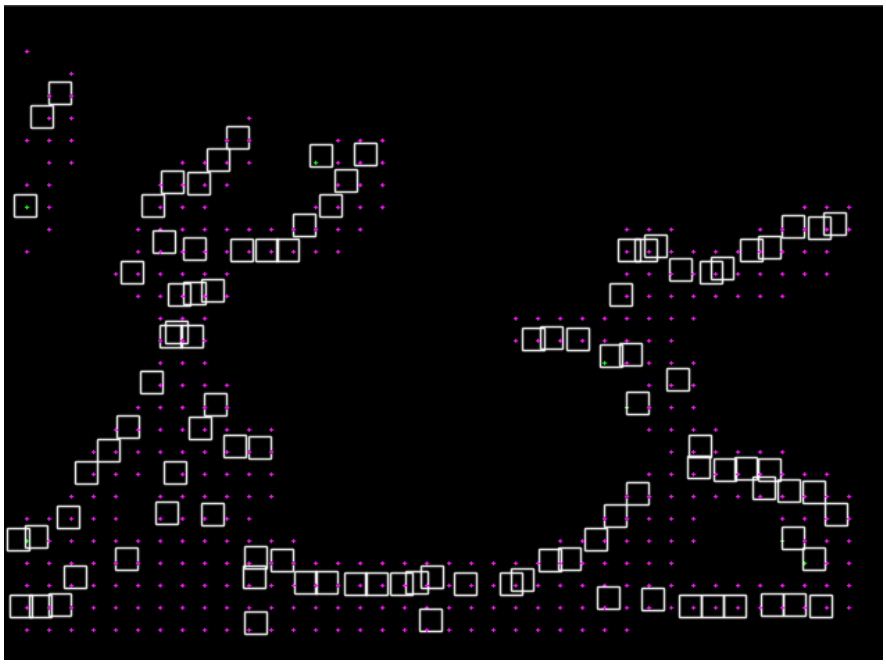


FIGURE 49 – On essais de valider des orientations locales par block et système de pointage (un block pointé est un block validé) pour ensuite tracer des polynôme dont la courbe serra interprétée

FIGURE 50 – On réalise les orientations locales avec poids sur les câbles, on valide les vecteurs qui sont référencés par d'autres (conservation d'une suite logique de points) puis on les relies. Plus tard, on éliminera les bords de carton avec la méthode de Hough (détection de lignes) puis on essaiera de réaliser une interpolation des points de vecteurs (trouver une courbe). Les résultats sont moyens pour les câbles blancs (faute du gradient).

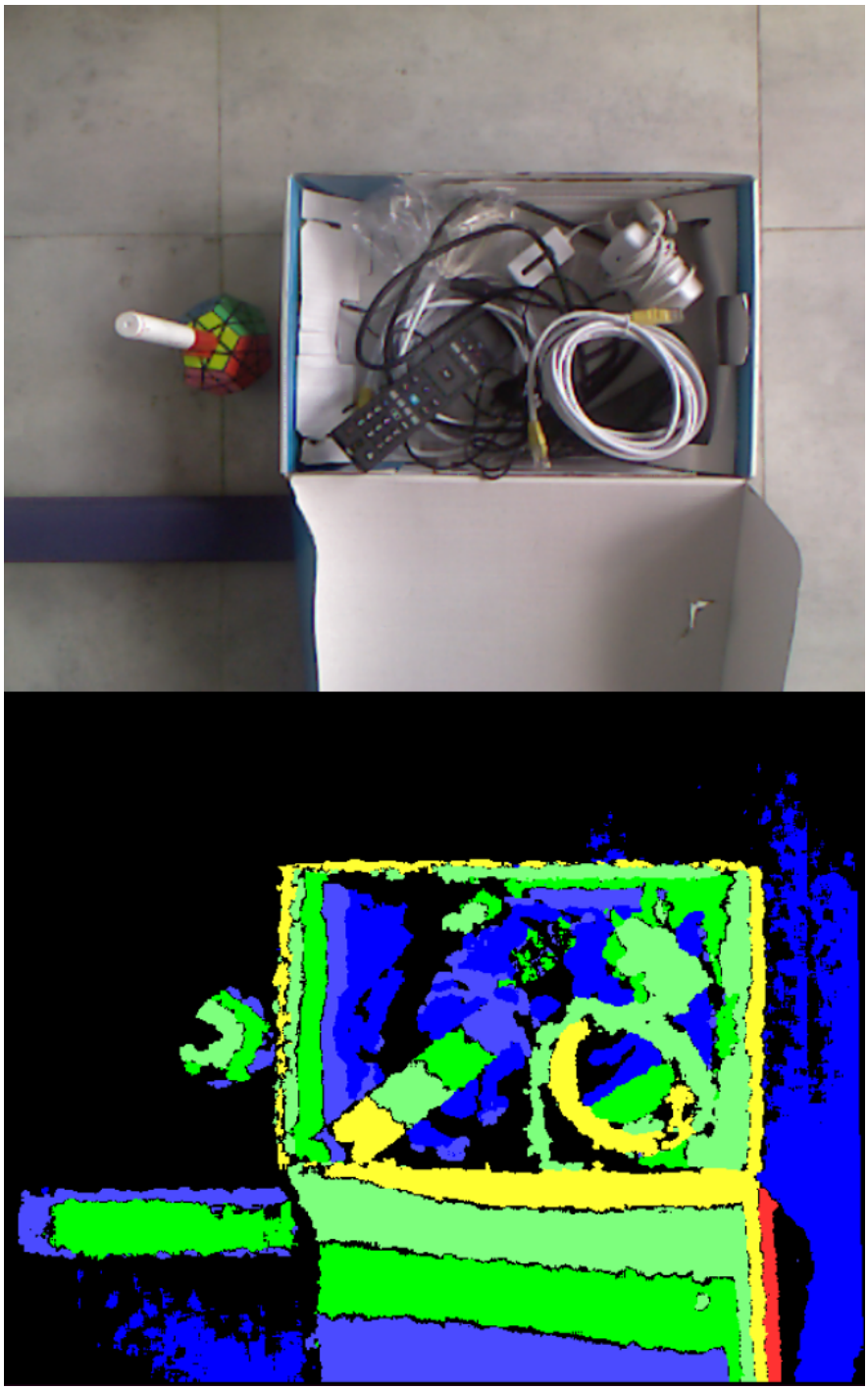


FIGURE 51 – Depuis les données de profondeur acquise par la Kinect et en passant par OpenNi, on segmente naïvement la scène complexe par profondeur en 6 calques et on la représente grâce à openCV. On va devoir approcher la problématique par propagation et détection de contours pour segmenter réellement l'objet (voir télécommande).

Annexe 4 : Réseaux de neurones

Listing 1 – On peux constater qu'au cour de l'apprentissage d'un réseau de neurones (ici, inception-v3) la précision de la reconnaissance n'augmente pas de façons linéaire

```
2016-05-11 16 :44 :00.583465 : Step 0 : Train accuracy = 53.0%
2016-05-11 16 :44 :00.583644 : Step 0 : Cross entropy = 1.298453
2016-05-11 16 :44 :00.772920 : Step 0 : Validation accuracy = 45.0%
2016-05-11 18 :43 :19.531381 : Step 10 : Train accuracy = 89.0%
2016-05-11 18 :43 :19.531540 : Step 10 : Cross entropy = 0.886562
2016-05-11 18 :43 :19.623894 : Step 10 : Validation accuracy = 66.0%
2016-05-11 20 :41 :49.434383 : Step 20 : Train accuracy = 79.0%
2016-05-11 20 :41 :49.434692 : Step 20 : Cross entropy = 0.787742
2016-05-11 20 :41 :49.526926 : Step 20 : Validation accuracy = 71.0%
2016-05-11 22 :40 :21.244895 : Step 30 : Train accuracy = 91.0%
2016-05-11 22 :40 :21.245044 : Step 30 : Cross entropy = 0.554185
2016-05-11 22 :40 :21.335518 : Step 30 : Validation accuracy = 83.0%
2016-05-12 00 :38 :51.286143 : Step 40 : Train accuracy = 89.0%
2016-05-12 00 :38 :51.286298 : Step 40 : Cross entropy = 0.552700
2016-05-12 00 :38 :51.375001 : Step 40 : Validation accuracy = 76.0%
2016-05-12 02 :37 :25.099880 : Step 50 : Train accuracy = 90.0%
2016-05-12 02 :37 :25.100035 : Step 50 : Cross entropy = 0.464012
2016-05-12 02 :37 :25.189902 : Step 50 : Validation accuracy = 80.0%
2016-05-12 04 :35 :58.498156 : Step 60 : Train accuracy = 91.0%
2016-05-12 04 :35 :58.498307 : Step 60 : Cross entropy = 0.472601
2016-05-12 04 :35 :58.591728 : Step 60 : Validation accuracy = 73.0%
2016-05-12 06 :34 :35.663803 : Step 70 : Train accuracy = 92.0%
2016-05-12 06 :34 :35.663948 : Step 70 : Cross entropy = 0.393777
2016-05-12 06 :34 :35.754954 : Step 70 : Validation accuracy = 82.0%
2016-05-12 08 :33 :20.634873 : Step 80 : Train accuracy = 91.0%
2016-05-12 08 :33 :20.635023 : Step 80 : Cross entropy = 0.402900
2016-05-12 08 :33 :20.725229 : Step 80 : Validation accuracy = 76.0%
```

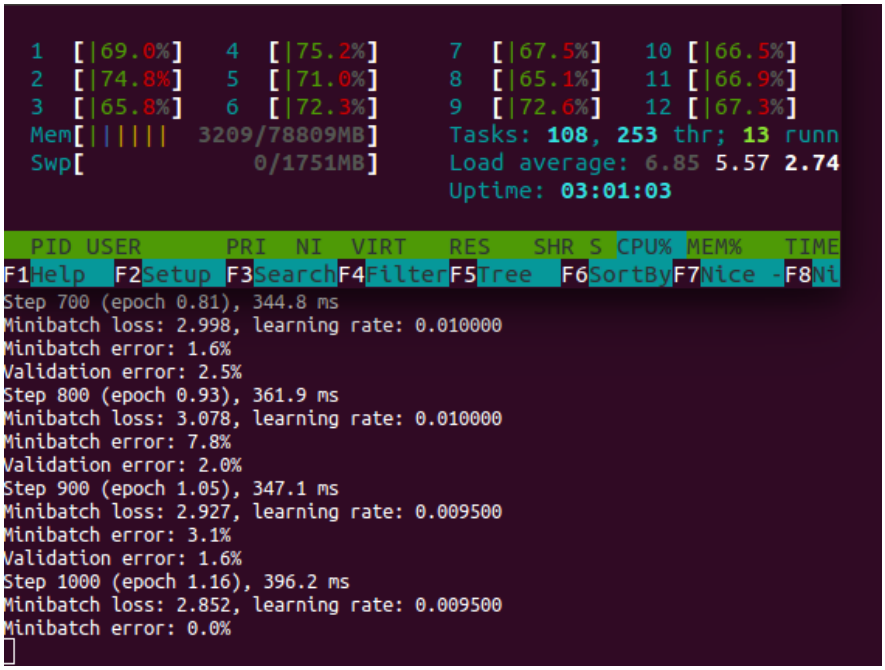
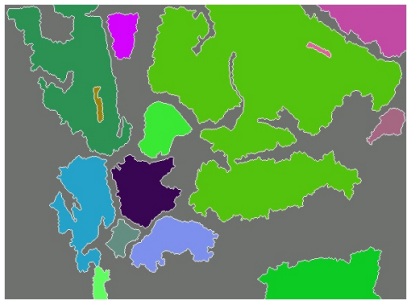


FIGURE 52 – Entrainement court d'un réseau de neurones convolutif sur MNIST (base d'images de chiffres manuscrits) avec tensorflow sur le serveur

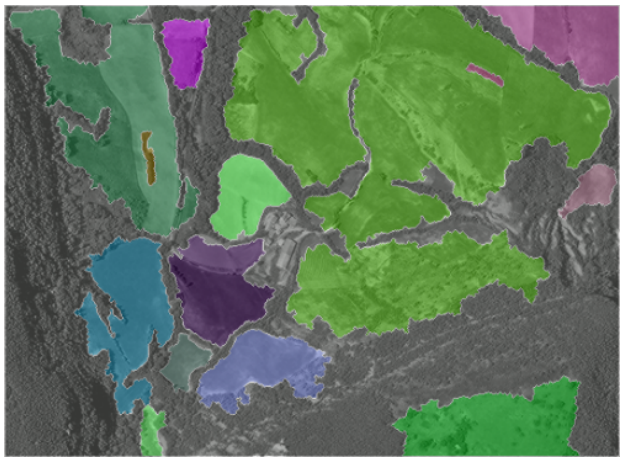
Annexe 5 : Segmentation par propagation de labels

(a) Image d'origine

(b) Supervision des labels



(c) Watershed



(d) Source/Segmentation

(e) Masque binaire obtenu

FIGURE 53 – Autre exemple de segmentation par propagation de labels

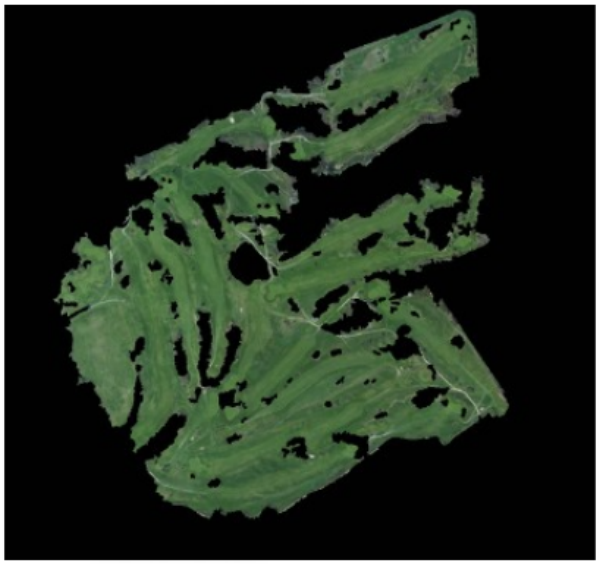
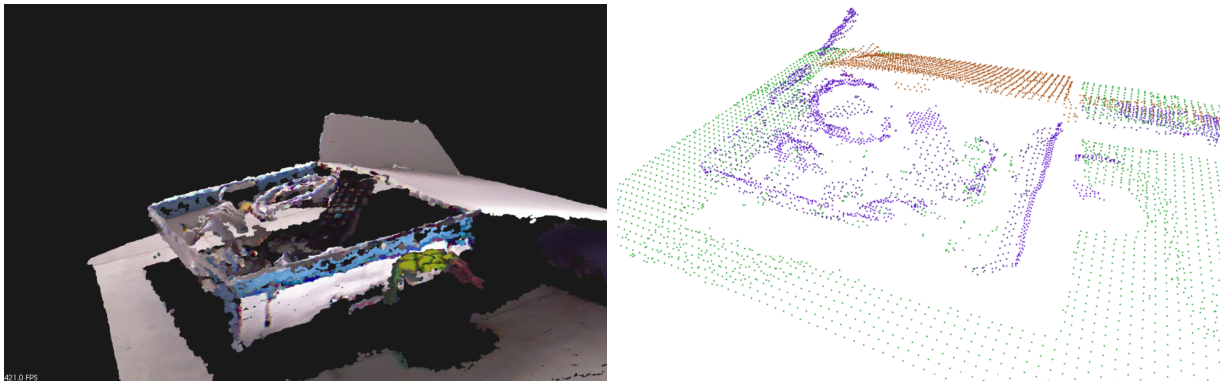


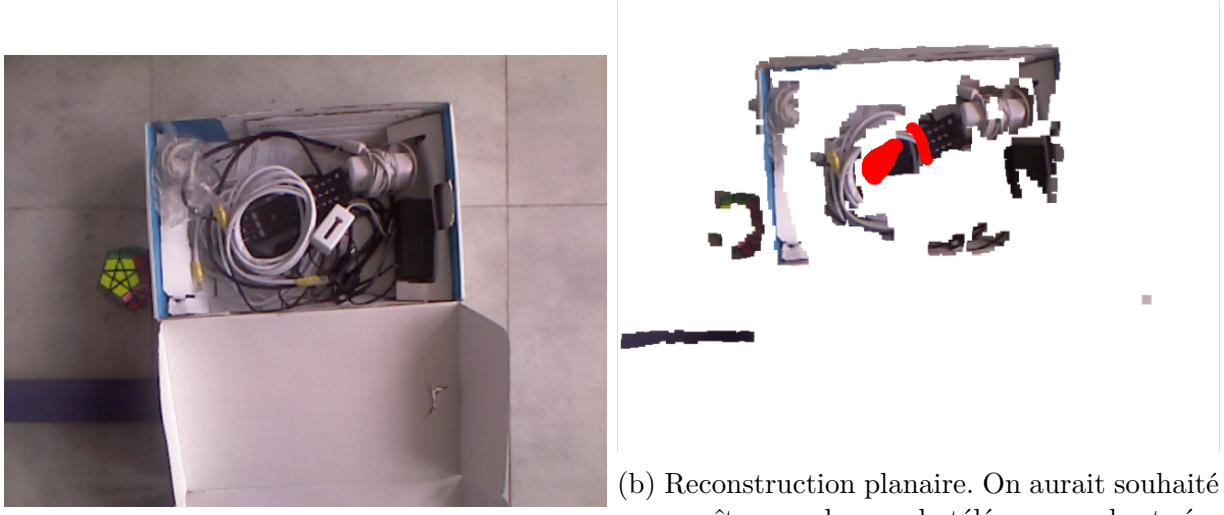
FIGURE 54 – Utilisation de l'implémentation et résultat après sélection de l'espace de liberté

Annexe 6 : Segmentation planaire de profondeur



(a) Visualisation (b) Segmentation

FIGURE 55 – Segmentation des plans de la profondeur



(a) Référence

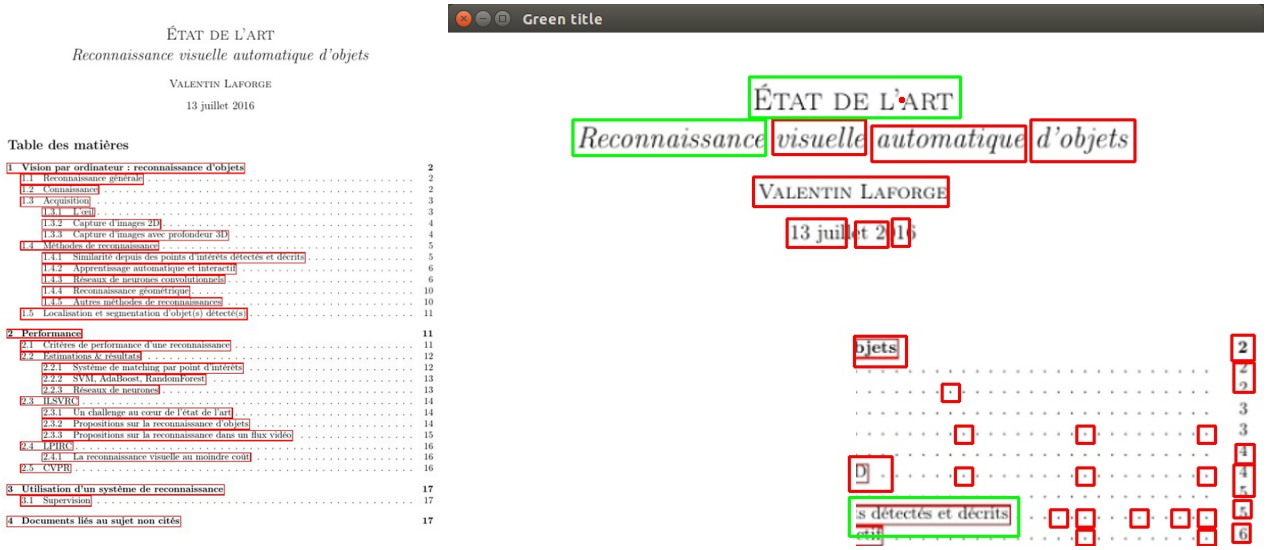
(b) Reconstruction planaire. On aurait souhaité reconnaître un plan sur la télécommande et réaliser de l'inpainting sur les région occultée, en rouge.

FIGURE 56 – Limites de l'implémentation

Annexe 7 : Extraction de Structure pour la Reconnaissance Visuelle d'Elements dans des Documents

Listing 2 – Exemple d'initialisation de la description des éléments structurants d'un document avec le segmenter basique de SERVED

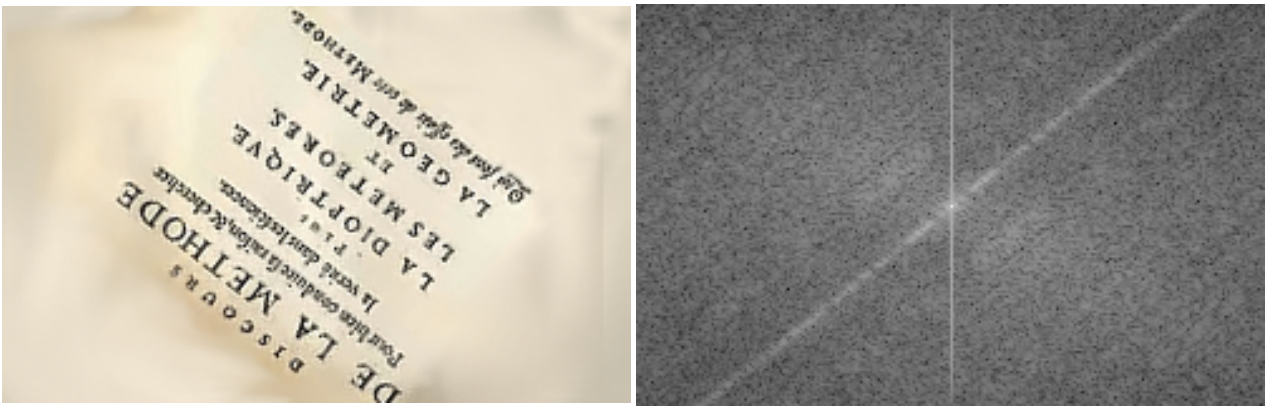

```
{"sample1":[
{"type":"undefined","value":"Undefined","bounding-box":[[71.0,770.0],[135.0,835.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[351.0,769.0],[656.0,828.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[187.0,760.0],[339.0,814.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[351.0,726.0],[655.0,763.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[406.0,698.0],[602.0,719.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[35.0,693.0],[177.0,773.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[187.0,655.0],[339.0,758.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[44.0,647.0],[171.0,686.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[356.0,601.0],[505.0,661.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[35.0,593.0],[177.0,629.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[187.0,588.0],[339.0,653.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[188.0,542.0],[335.0,586.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[516.0,539.0],[656.0,651.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[353.0,536.0],[505.0,604.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[35.0,513.0],[177.0,591.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[518.0,505.0],[651.0,528.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[564.0,484.0],[655.0,504.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[45.0,459.0],[168.0,500.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[189.0,457.0],[338.0,530.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[353.0,455.0],[505.0,534.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[359.0,438.0],[503.0,455.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[521.0,394.0],[650.0,475.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[361.0,387.0],[499.0,433.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[120.0,341.0],[334.0,356.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[39.0,331.0],[336.0,437.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[470.0,320.0],[664.0,378.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[124.0,300.0],[260.0,321.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[353.0,275.0],[654.0,367.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[135.0,275.0],[338.0,296.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[461.0,242.0],[515.0,269.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[579.0,237.0],[658.0,274.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[517.0,237.0],[576.0,270.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[396.0,237.0],[459.0,269.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[250.0,237.0],[314.0,269.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[143.0,237.0],[217.0,270.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[107.0,237.0],[140.0,269.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[316.0,236.0],[392.0,269.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[220.0,236.0],[247.0,269.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[36.0,235.0],[104.0,270.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[412.0,140.0],[661.0,182.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[635.0,128.0],[664.0,155.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[95.0,128.0],[142.0,156.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[37.0,116.0],[658.0,226.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[412.0,114.0],[478.0,143.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[35.0,113.0],[103.0,144.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[469.0,112.0],[621.0,145.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[154.0,112.0],[280.0,179.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[93.0,58.0],[163.0,86.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[587.0,32.0],[649.0,95.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[192.0,23.0],[508.0,113.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[233.0,45.0],[358.0,79.0]]},{"type":"undefined","value":"Undefined","bounding-box":[[510.0,14.0],[661.0,95.0]]},
{"type":"undefined","value":"Undefined","bounding-box":[[35.0,14.0],[186.0,106.0]]}]}
```



(a) Document original

(b) Pré-traitement

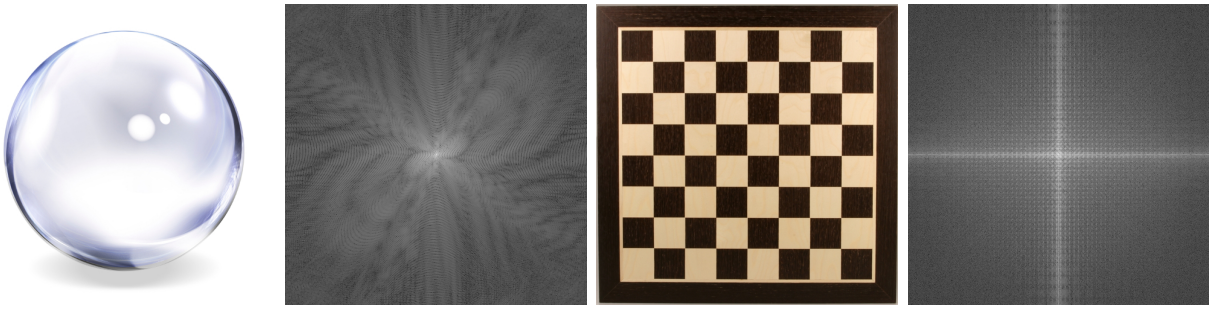
FIGURE 57 – Limites de l'extraction de titre (police peu compacte), titre obtenu à confiance minimum 55% :
ETAT DE L'ART-Reconnaissance



(a) Image originale

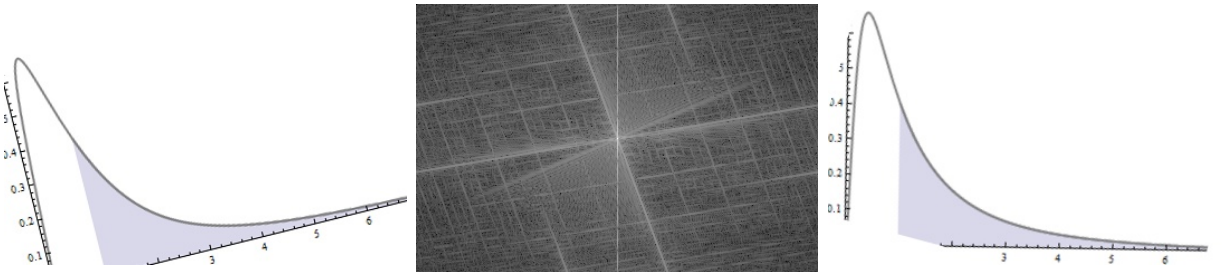
(b) DFT

FIGURE 58 – Orientation d'une image depuis sa transformée de Fourier discrète où le nombre d'axes obtenus est restreint



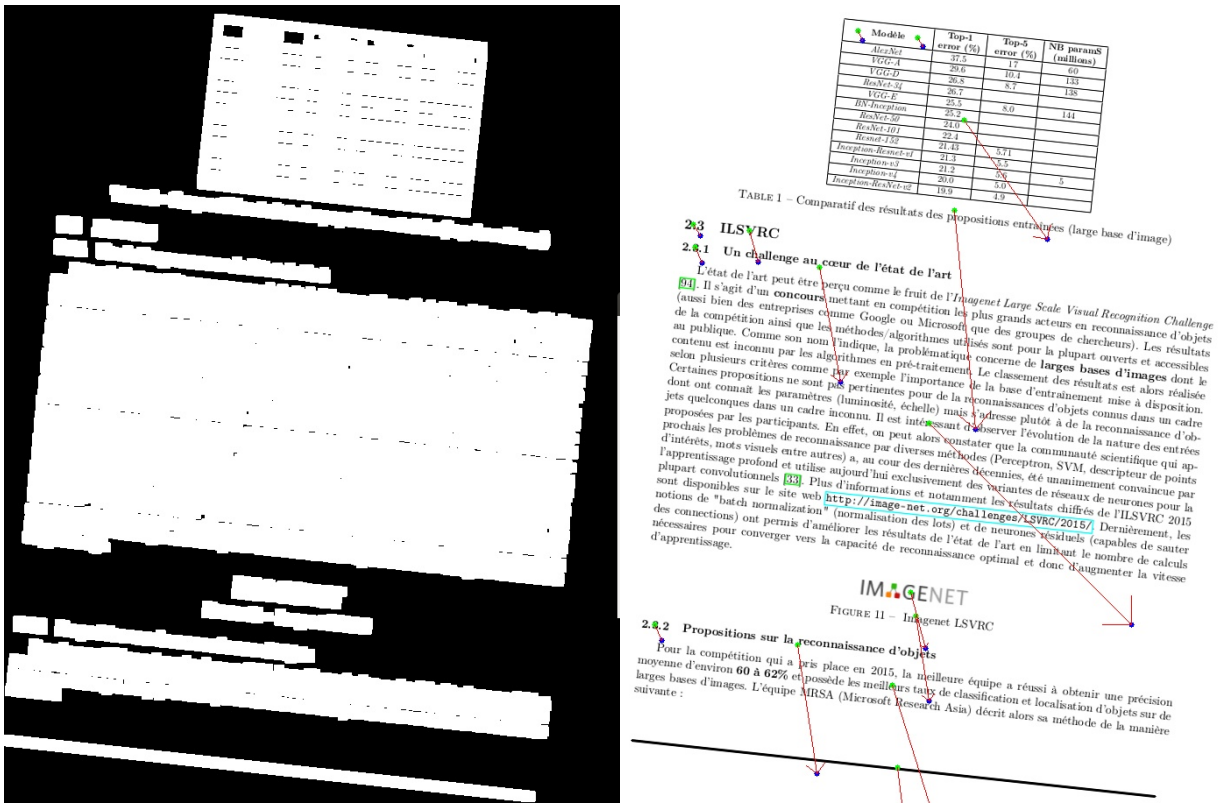
(a) Elément circulaire (b) DFT "circulaire" (c) Grille (d) DFT "grille"

FIGURE 59 – TFD sur cas particuliers



(a) Image originale (b) TFD particulière (grille) (c) Réalignement automatique

FIGURE 60 – Réaction de la TFD sur un graphique

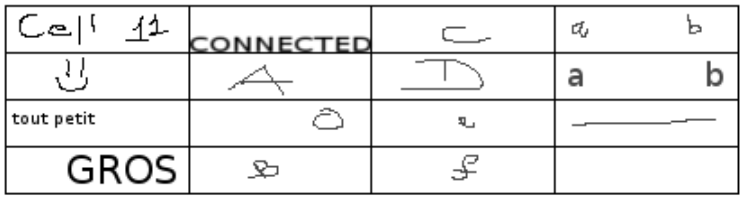


(a) Blocs binaires

(b) Orientation depuis les moments des blocs

FIGURE 61 – Calcul des angles des blocs en utilisant les moments des images (approche non retenue de l'estimation de l'angle d'un document)

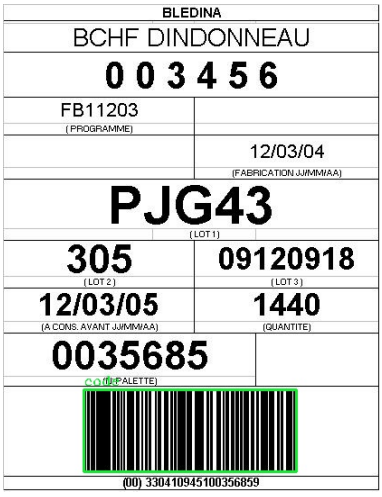
cell 1	cell 2	cell 3	cell 4	cell 5
cell 6	cell 7	cell 8	cell 9	cell 10



cell 1 {94.8955}	cell 2 {94.07323}	cell 3 {90.49428}	cell 4 {91.62295}	cell 5 {92.67742}
cell 6 {91.40312}	cell 7 {92.444824}	cell 8 {90.91837}	cell 9 {90.96774}	cell 10 {94.926704}

Ce.“ ii {56.855736}	CONNECTED {83.77768}	C, {59.366608}	:1, b{ 68.40711}
kl!) {43.108845}	Ar {56.432907}	“m {55.097294}	a b {90.643074}
tout p etit {77.84977}	C {66.709335}	L {72.13845}	
GROS {87.81193}	b {55.146065}	:5 ? {66.4209}	

FIGURE 63 & TABLE 1 – Extraction de tableau avec la confiance correspondante



(a)

(b)

FIGURE 64 – Exemples de résultat d'une approche détection de code barre basée sur l'étude d'un modèle de lignes parallèles et ordonnées sur des critères d'espacements relatifs. Non retenue car peu efficace et trop sensible.