Évaluation de l'interprétation d'images

Baptiste Hemery

UNIVERSITÉ de CAEN/BASSE-NORMANDIE
U.F.R. de Sciences
École doctorale S.I.M.E.M

THÈSE

présentée par

M. Baptiste Hemery

et soutenue

le 2 décembre 2009

en vue de l'obtention du

Doctorat de l'Université de Caen Basse-Normandie
Spécialité : Traitement du signal et des images

Arrêté du 7 août 2006

Évaluation de l'interprétation d'images

Membre du Jury

Philippe Bolon	Professeur des universités	Laboratoire LISTIC, Polytech'Savoie	(Rapporteur)
François Brémond	Chargé de Recherche HDR	INRIA, Sophia Antipolis	(Rapporteur)
Ludovic Macaire	Professeur des universités	Laboratoire LAGIS, Université des Sciences et Technologies de Lille	
Fréderic Jurie	Professeur des universités	Laboratoire GREYC, Université de Caen Basse Normandie	
Hélène Laurent	Maître de conférences	Institut PRISME, ENSI de Bourges	
Christophe Rosenberger	Professeur des universités	Laboratoire GREYC, ENSICAEN	(Directeur de thèse)

Pour Jacques

Remerciements

Je tiens tout d'abord à remercier le ministère de l'éducation nationale pour son soutien financier qui m'a permis de mener à bien ce travail au cours de ces trois dernières années.

Je remercie également monsieur Philippe Bolon, professeur à Polytech'Savoie, et monsieur François Brémond, chargé de recherche CNRS HDR à l'INRIA, pour avoir accepté de rapporter ce manuscrit de thèse. Je tiens à remercier monsieur Ludovic Macaire, professeur à l'université de Lille, et monsieur Frédéric Jurie, professeur à l'université de Caen, pour avoir accepté d'examiner ce travail.

Au cours de ma thèse, j'ai eu l'occasion de travailler durant mes deux premières années au sein du LVR [1], intégré depuis 2008 au sein de l'institut PRISME [2], et , au sein du laboratoire GREYC [3] pour ma troisième année. Je remercie donc Youssoufi Touré, directeur du LVR puis de l'institut PRISME. Je remercie également Etienne Grandjean, directeur du GREYC qui m'a accueilli pour ma troisième année de thèse. Je remercie également Luc Brun de m'avoir accepté dans l'équipe Image du GREYC.

Ce changement de laboratoire au cours de la troisième année fut possible grâce à l'école doctorale SIMEM [4], à Caen, qui m'a accueilli et je l'en remercie. Je remercie également mon école doctorale d'origine, l'EDST [5] à Orléans.

Je remercie tout particulièrement ceux par qui ces trois années ont été possibles. Je remercie donc Christophe Rosenberger, mon directeur de thèse, ainsi que Hélène

1. Laboratoire Vision et Robotique
2. Institut Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique et Energétique
3. Groupe de REcherche en Informatique, Image, Automatique et Instrumentation de Caen
4. Structure, Information, Matière et Matériaux
5. École Doctorale Sciences et Technologie

Laurent qui a co-encadré ma thèse. Je remercie également Bruno Emile qui a suivi ma thèse avec intérêt. Je les remercie pour leur professionnalisme ainsi que l'amitié qu'ils m'ont portée.

Enfin, je remercie tous ceux qui m'ont apporté leurs amitiés au cours des trois années de ce travail, notamment mes collègues de bureau Yannick, Adel, Hazem, Romain et Mohammad. J'ai également apprécié l'accueil chaleureux des membres du DRI [6] de l'Ensicaen et de l'équipe SISTEM [7] du GREYC.

Finalement, je remercie mes proches qui m'ont toujours encouragé et soutenu, notamment Annick, ma mère, ainsi que Chrystel qui partage aujourd'hui ma vie. Je dédie ce travail à Jacques, mon père, parti trop tôt.

6. Département des Relations Industrielles
7. Sécurité Informatique, Sécurité des Transactions Electroniques et Monétique

Résumé

Les algorithmes de traitement d'images regroupent un ensemble de méthodes qui vont traiter l'image depuis son acquisition par un capteur (webcam, satellite, échographe...) jusqu'à l'extraction de l'information utile pour une application donnée (détection d'un objet particulier, mesure quantitative...). Parmi ces algorithmes, certains ont pour but de détecter, localiser et reconnaître un ou plusieurs objets dans une image. Compte tenu des enjeux liés à l'extraction de ces informations, notamment pour des applications dans le domaine militaire ou médical, il est particulièrement important que les résultats fournis par les algorithmes d'interprétation d'images soient les plus pertinents possible. Le problème traité dans cette thèse réside dans l'évaluation de résultats d'interprétation d'une image ou une vidéo lorsque l'on dispose de la vérité terrain associée. Les enjeux sont nombreux comme la comparaison d'algorithmes d'interprétation pour une application particulière, l'évaluation d'un algorithme au cours de son développement ou son paramétrage optimal. Il existe deux étapes majeures en interprétation d'images à savoir la localisation et la reconnaissance d'objets. De nombreuses méthodes et métriques ont été proposées dans la littérature afin d'évaluer un résultat de localisation ou de reconnaissance notamment dans le cadre de compétitions (Technovision, Pascal...) ou de conférences (PETS, ECCV...). Il reste cependant difficile d'estimer la pertinence d'une métrique et encore plus de déterminer celle qui doit être préconisée.

Nous proposons dans cette thèse une formalisation des propriétés attendues d'une métrique de localisation. Nous réalisons une étude comparative rigoureuse des métriques de localisation de l'état de l'art au vu de ces propriétés. Cette étude a permis une caractérisation précise du comportement des métriques. Nous réalisons un travail similaire sur les méthodes de reconnaissance utilisant une représentation locale des objets dans le but de quantifier une erreur de reconnaissance : un résultat aboutissant à une reconnaissance erronée d'un objet comme un objet de la classe

« chat » au lieu de « chien » doit, par exemple, être considéré comme meilleur par rapport à celui affectant le même objet à la classe « voiture ».

Nous avons également mis au point une méthode d'évaluation d'un résultat d'interprétation d'une image exploitant les leçons de ces études comparatives. L'avantage de la méthode proposée est de pouvoir évaluer un résultat d'interprétation d'une image en prenant en compte à la fois la qualité de la localisation, de la reconnaissance et de la détection de la présence d'objets d'intérêt dans l'image. Un appariement est réalisé entre la vérité terrain et le résultat d'interprétation à évaluer. Chaque objet apparié contribue au score global du résultat en prenant en compte les erreurs de localisation et de reconnaissance. Le score global tient compte enfin de la sur-détection et la sous-détection des objets. Le comportement de cette méthode d'évaluation a été testé sur une large base de test (issue de la base PASCAL) et présente des résultats intéressants. Tout en respectant les propriétés utilisées lors de l'étude comparative, cette nouvelle méthode permet, grâce à plusieurs paramètres, d'obtenir un comportement adapté à l'application visée en prenant en compte notamment, le mode de mise en correspondance utilisé ou bien l'importance relative de la localisation et de la reconnaissance dans le calcul du score global.

Summary

Image processing algorithms include a set of methods that process the image from its acquisition by a sensor (camera, satellite, ultrasound ...) to the extraction of useful information for a given application (detection of a particular object, quantitative measure ...). Among these algorithms, some are dedicated to detect, locate and identify one or more objects in an image. Given the challenges associated with extracting this information (including military or medical applications), it is particularly important that results provided by image interpretation algorithms are as relevant as possible. The problem addressed in this thesis is the evaluation of interpretation results of an image or a video, given the associated ground truth. The challenges are multiple such as the comparison of algorithms for a particular application, the evaluation of an algorithm during its development or its optimal setting. There are two major steps in image interpretation which are the localization and recognition. Many methods and metrics have been proposed in the literature to evaluate localization or recognition results in several competitions (Technovision, Pascal ...) or conferences (PETS, ECCV ...). However, it remains difficult to estimate the relevance of a metric and even to determine which should be advocated for a given application.

We propose in this thesis a formalization of the expected properties of a localization metric. We perform a rigorous comparative study of localization metrics of the state from the art considering these properties. This study has allowed a precise characterization of the metrics behavior. We perform a similar work on recognition methods using a local representation of objects in order to quantify a recognition error : a result leading to an erroneous object recognition as an object of class « cat » instead of class « dog » must, for example, be considered better compared to the same object affected to the class « car ».

We have also developed an evaluation method of an image interpretation result making use of the lessons from these comparative studies. The advantage of the proposed method is to evaluate an image interpretation result taking into account both the quality of localization, recognition and detection of objects of interest in the image. A matching process between the ground truth and the evaluated interpretation result is realized. Each matched object contributes to the overall score by taking into account localization and recognition errors. The overall score takes into account over-detection and under-detection of objects. The behavior of this method was tested on a benchmark (from the PASCAL database) and presents some interesting results. While respecting the properties used in the comparative study, this new method, thanks to several settings, obtains an appropriate behavior in the intended application, taking into particular account the matching mode or the relative importance of localization and recognition in the computation of the overall score.

Table des matières

Introduction

« If you can not measure it, you can not improve it. »

Sir William Thomas Kelvin

Positionnement de la problématique

La vidéo-surveillance a connu un très grand développement au cours des dix dernières années. Les premières caméras de vidéo-surveillance ont été introduites à Londres en 1989, ce sont aujourd'hui plus de *500 000* caméras qui surveillent les rues londoniennes. Plus de 4 millions de caméras ont été réparties dans le Royaume-Uni. En France, *340 000* caméras ont été installées en 2007 tandis qu'un million sont prévues pour 2009. Avec un tel accroissement de la vidéo-surveillance, le développement d'algorithmes de traitement automatique des images pour la détection d'évènements anormaux devient fondamentale pour assurer la sécurité dans les rues, les moyens de transport, les banques ou les magasins.

Dans un tout autre domaine, l'imagerie médicale a également connu un très large essor. Les premières images médicales remontent à plus d'un siècle avec l'apparition de la radiographie. On dispose aujourd'hui d'un grand nombre d'appareils d'imagerie médicale tels que l'imagerie par résonance magnétique (IRM), l'échographie ou la tomographie. Ainsi, ce sont plus de deux millions d'examens IRM qui sont réalisés en France chaque année, et plus de 38 millions aux Etats-Unis. De plus, les données obtenues par ces appareils sont de plus en plus conséquentes. Dans ce domaine également, il est nécessaire d'avoir recours à des algorithmes afin d'extraire automatiquement l'information pertinente et d'aider au diagnostic.

A partir de ces deux exemples, nous pouvons voir que l'extraction automatisée d'informations pertinentes sur les objets présents dans des images a un intérêt

grandissant. Cette opération qui consiste à détecter, localiser et/ou reconnaître un ou plusieurs objets dans une image est une étape importante pour l'*interprétation d'images*. Les domaines d'application de l'interprétation d'images sont très variés de par la grande diversité des capteurs d'images disponibles : satellite, vidéo-surveillance, webcam, échographe... Avec un satellite par exemple, les applications vont concerner la création de cartes géographiques, la mesure d'exploitations agricoles ou bien le suivi climatique. Lorsque le capteur est une caméra, les applications vont de la biométrie au contrôle qualité de pièces industrielles, en passant par la détection automatique des visages afin de ne plus jamais rater ses photos... Compte tenu des enjeux liés à l'extraction de ces informations, il est particulièrement important que les résultats fournis par les algorithmes d'interprétation d'images soient les plus pertinents possible. Cependant, comme le dit Sir William Thomas Kelvin, comment peut-on améliorer la qualité de ces résultats si nous ne pouvons la mesurer ?

Objectifs

La problématique traitée dans cette thèse réside dans l'évaluation de résultats d'interprétation d'une image ou d'une vidéo lorsque l'on connaît la vérité terrain associée. Les enjeux sont nombreux comme la comparaison d'algorithmes d'interprétation pour une application particulière, l'évaluation d'un algorithme au cours de son développement ou son paramétrage optimal. Ainsi, l'objectif est de permettre la mesure de la qualité des résultats d'interprétation d'images afin de l'améliorer.

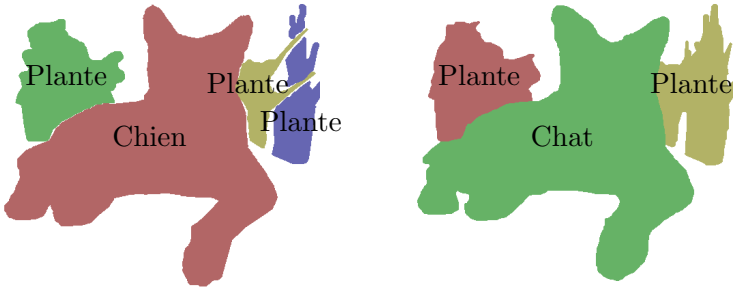
Concrètement, nous devons être capables de déterminer la qualité des résultats fournis par un algorithme d'interprétation en se basant sur une vérité terrain. Un exemple d'une image, provenant de la base de données Caltech256 [1], ainsi que de la vérité terrain associée est présenté à la figure 0.1.

Étant donné la multiplicité des résultats possibles fournis par un algorithme d'interprétation, nous nous intéressons ici à trois résultats : les résultats de localisation, les résultats de reconnaissance et, enfin, les résultats combinant à la fois localisation et reconnaissance. La figure 0.2 présente quatre résultats d'interprétation contenant à la fois la localisation et la reconnaissance des objets présents dans la scène. Nous pouvons voir qu'aucun résultat ne correspond parfaitement à la vérité terrain. Certains objets n'ont pas du tout été détectés dans deux résultats d'interprétation. Certains présentent des altérations de reconnaissance, d'autres des problèmes de détection. Enfin, la localisation est toujours un problème, bien qu'il soit moins important. Notre objectif est donc de déterminer automatiquement, à partir de la

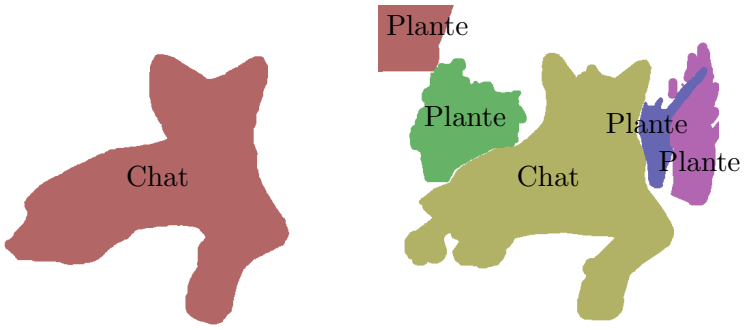
(a) Image originale (b) Vérité terrain

FIGURE 0.1 – Une image et sa vérité terrain associée

vérité terrain présente à la figure 0.1, le meilleur résultat d'interprétation.



(a) Résultat 1 (b) Résultat 2



(c) Résultat 3 (d) Résultat 4

FIGURE 0.2 – Quatre résultats d'interprétation de l'image de la figure 0.1

Organisation du manuscrit

Ce manuscrit de thèse est composé de cinq parties :

- Le premier chapitre formalise les différents algorithmes d'interprétation d'images et en présente quelques exemples,
- Le second chapitre est un état de l'art des méthodes permettant d'évaluer les algorithmes d'interprétation. Nous y présentons tout d'abord les représentations utilisées pour les résultats de localisation et de reconnaissance. Nous y présentons ensuite les métriques permettant d'évaluer les résultats de localisation, les résultats de reconnaissance ainsi que les méthodes permettant d'évaluer un algorithme dans son ensemble,
- Le troisième chapitre présente trois études comparatives qui nous ont permis de mettre en avant certaines métriques et méthodes provenant de l'état de l'art. Une première étude compare les métriques d'évaluation de résultats de localisation, tandis que les deux suivantes concernent la reconnaissance,
- Le quatrième chapitre présente la méthode d'évaluation globale que nous avons développé. Pour cela, nous avons utilisé les résultats des études comparatives du chapitre précédent,
- Enfin, une conclusion et des perspectives viennent clore ce manuscrit de thèse.

Chapitre 1

Interprétation d'images

Ce chapitre présente le contexte de l'interprétation d'images. Nous proposons une formalisation des différents algorithmes ayant pour vocation la localisation et la reconnaissance d'objets d'intérêt dans une image. Nous illustrons ces algorithmes par des cas concrets de l'état de l'art.

Sommaire

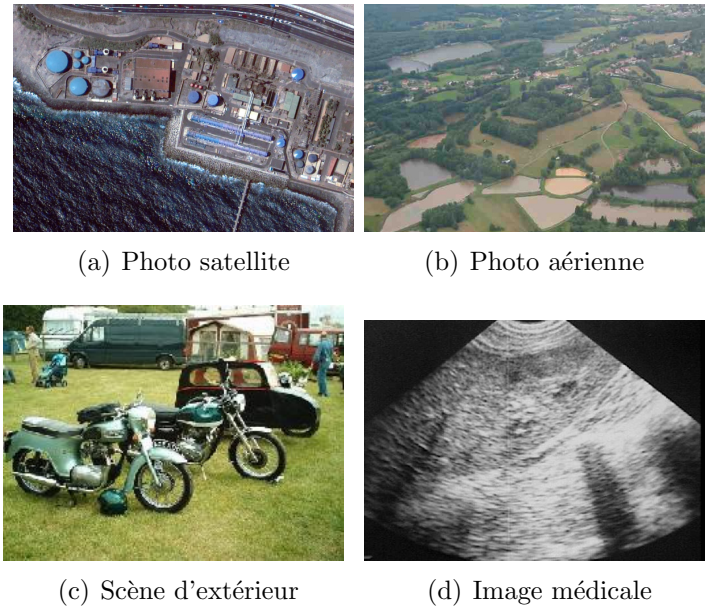
1.1 Introduction

Le traitement d'images est un domaine vaste. De nombreux algorithmes permettent de traiter les images ou la vidéo depuis leur acquisition par un capteur (appareil photographique, webcam, satellite, échographe...) jusqu'à l'extraction de l'information utile pour l'application (détection d'un objet particulier, mesure quantitative...). L'extraction d'un maximum d'informations sur les objets composant une scène, le plus automatiquement possible, est l'objectif principal de l'interprétation d'images. Les algorithmes d'interprétation sont donc des algorithmes de haut niveau et occupent une place importante dans le processus de traitement.

Une image correspond à la représentation d'une scène réelle sur un support. Ce support peut être une toile ou bien du papier lorsque l'on s'intéresse à la peinture

ou bien à la photographie. Cependant, dans le cadre du traitement automatisé des images, c'est-à-dire dans le domaine informatique, ce support prend généralement la forme d'une matrice dont les valeurs correspondent à la numérisation d'un signal acquis par un capteur. Ce capteur est sensible à un rayonnement physique et peut renvoyer une ou plusieurs données correspondant chacune à une longueur d'onde (lumière visible, infrarouge, rayons X...).

Les capteurs permettant d'acquérir des images numériques étant de différentes natures, les images que nous traitons le sont également. Nous pouvons obtenir des images dans le domaine visible prises depuis un satellite ou bien au sol, mais nous pouvons également obtenir des images dans des domaines non visibles par l'homme, tels que les infrarouges ou bien les rayons X. Quelques images, provenant de différents systèmes d'acquisition, sont présentées à la figure 1.1.



(a) Photo satellite (b) Photo aérienne

(c) Scène d'extérieur (d) Image médicale

FIGURE 1.1 – Exemples d'images pouvant être traitées en interprétation d'images

Par conséquent, les algorithmes d'interprétation d'images peuvent être utilisés pour de nombreuses applications : détection de la présence d'objets, comptage, contrôle qualité, localisation dans l'image... Pour chacune de ces applications spécifiques, il existe plusieurs algorithmes permettant de traiter le problème. Il est alors nécessaire de définir clairement leurs principes de fonctionnement, notamment les données qu'ils utilisent en entrée ou qu'ils renvoient, afin de pouvoir les évaluer

correctement.

Dans ce chapitre, les principaux types d'algorithmes d'interprétation d'images sont définis et quelques exemples sont présentés. Une discussion sur les algorithmes d'interprétation d'images est enfin proposée.

1.2 Formalisation de l'interprétation d'images

Parmi les différents algorithmes d'interprétation d'images, on peut distinguer trois types principaux. Certains algorithmes présentent un résultat sous la forme d'une image. C'est principalement le cas des algorithmes de localisation dont le résultat est le plus souvent une image contenant la frontière des objets localisés par exemple. D'autres algorithmes renvoient une donnée textuelle ou alphanumérique (position d'un objet dans l'image, liste d'éléments présents, valeurs numériques, vecteurs...). C'est le cas des algorithmes dont le but est de prédire la classe d'un objet ou bien de compter le nombre d'objets présents dans une image. Enfin, des algorithmes hybrides vont renvoyer une image contenant des informations de type données textuelles. C'est le cas des algorithmes qui sont capables de localiser un objet dans une image tout en prédisant sa classe. La valeur des pixels de l'image renvoyée par ce genre d'algorithme va contenir les deux types d'informations : la localisation de l'objet et sa classe.

Si les principaux objectifs des algorithmes d'interprétation d'images sont la reconnaissance et la localisation d'objets d'intérêt dans une image, il existe d'autres algorithmes d'interprétation d'images tels que les algorithmes de détection, de suivi de cible ou bien ceux visant à faire du contrôle qualité sur des chaînes de fabrication. Ces algorithmes sont développés à partir des algorithmes de reconnaissance et de localisation et sont souvent définis spécifiquement pour une application.

Cette partie vise à décrire et à formaliser le fonctionnement de plusieurs algorithmes classiquement utilisés en interprétation d'images [2]. Des exemples d'algorithmes sont également présentés afin d'illustrer le propos par quelques cas concrets.

1.2.1 La chaîne de traitement d'images

Il y a deux points de vue dans l'interprétation d'une image : le premier consiste à voir cette étape comme le traitement de haut niveau dans la chaîne de traitement d'images, qui est le point de vue historique, et le deuxième comme un traitement à part entière devant être robuste à différents défauts sur l'image (dégradation de

l'image...).

La chaîne de traitement d'images, présentée dans la figure 1.2, regroupe les différents traitements qui peuvent être appliqués à une image avant de pouvoir l'interpréter. Nous pouvons voir qu'après la phase d'acquisition suit une phase de prétraitement. Cette phase sert à traiter les défauts de l'image dus à l'acquisition. Par exemple, un temps d'exposition trop court peut être corrigé en rehaussant le contraste de l'image. Cette phase n'est pas toujours nécessaire si les conditions d'acquisitions de l'image sont bonnes.

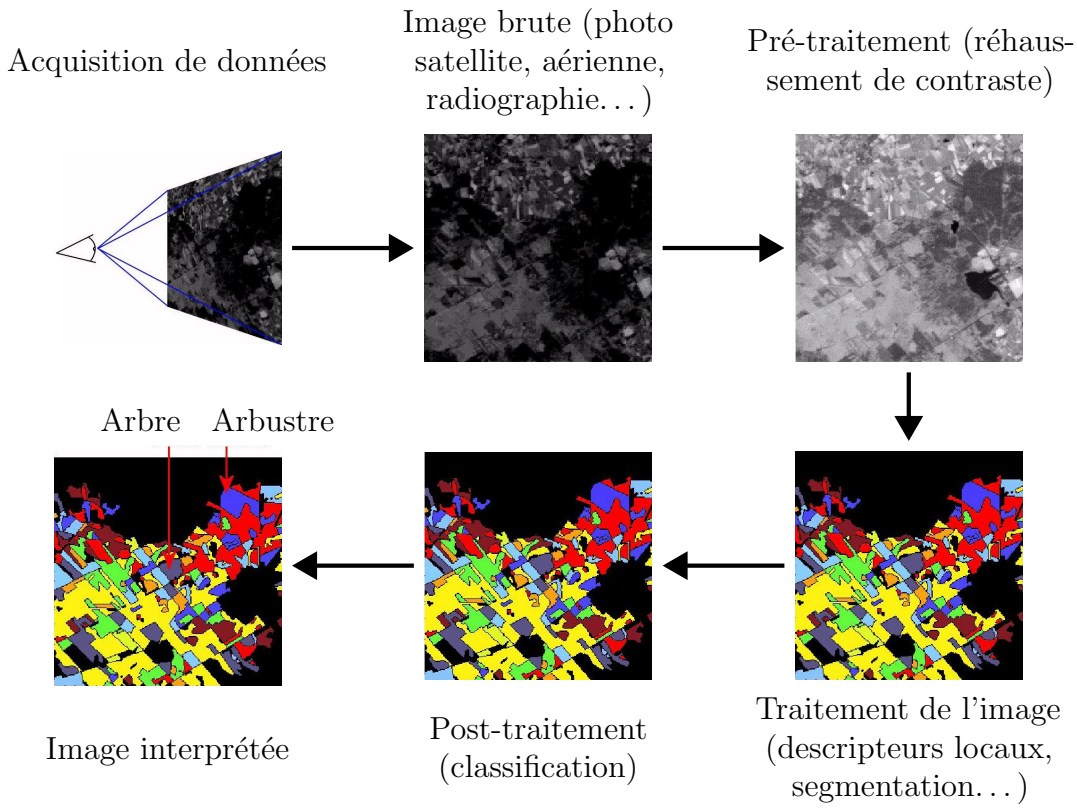


FIGURE 1.2 – Une chaîne de traitement d'une image (images du ©LASTI).

Suit alors une phase de traitement de l'image qui va permettre de la rendre interprétable. Cette phase comporte, selon l'algorithme d'interprétation utilisé, une phase de segmentation ou bien une extraction de données caractéristiques calculées à partir de descripteurs locaux. Vient ensuite une phase de post-traitement qui va permettre d'obtenir l'image interprétée, c'est à dire une image comportant l'information sur les objets de l'image. Cette phase utilise généralement des méthodes de classification

ou d'apprentissage afin d'interpréter les données extraites précédemment. L'interprétation d'images est donc la combinaison du traitement et du post-traitement des images, ce qui en fait le maillon final de la chaîne de traitement d'images.

Le second point de vue de l'interprétation d'images, plus actuelle, considère l'interprétation d'images comme un traitement à part entière. Ainsi, les images en entrée d'algorithmes ne subissent aucun pré-traitement. En effet, ces pré-traitement peuvent parfois détériorer l'image plus qu'ils ne l'améliorent. Ainsi, les artefacts de l'image, tels que le bruit ou la variation de luminosité, correspondant à des problèmes auxquels les algorithmes d'interprétation doivent être robustes, au lieu de reposer sur un pré-traitement qui corrigerait ces problèmes [3, 4, 5, 6].

1.2.2 Les algorithmes non supervisés de localisation

Les algorithmes non supervisés de localisation vont permettre de localiser des objets dans une image ou une vidéo sans avoir de connaissance *a priori* sur ces objets d'intérêt. Dans ce cas, l'algorithme n'a pas recours à l'utilisation d'une base de connaissances afin d'apprendre à détecter les objets. L'avantage est que l'algorithme va pouvoir localiser tout type d'objet dans les images, mais en contrepartie, il n'a aucune information sur le type d'objet qui a été localisé. Ces algorithmes sont essentiellement basés sur la segmentation ou la détection de mouvement dans des vidéos.

Formalisation du problème

L'objectif des algorithmes de localisation non supervisés est de donner une localisation d'un ou plusieurs objets comme résultat en ne prenant en compte qu'une ou plusieurs images comme entrée de l'algorithme.

$$Localisation : I \mapsto Z_i \tag{1.1}$$

avec I l'image d'entrée et Z_i le résultat de localisation.

Exemples

De nombreuses méthodes existent dans la littérature pour la localisation non supervisée, notamment d'objets en mouvement dans des vidéos [7, 8]. Une méthode simple consiste à faire une différence temporelle, c'est-à-dire à soustraire deux images prises à des temps différents. Cette méthode pose cependant quelques problèmes

lorsqu'il s'agit de détecter des objets non texturés.

Il est aussi possible de faire une estimation de l'avant-plan, c'est-à-dire le plan contenant les objets que l'on souhaite détecter, en soustrayant l'arrière plan, avec une image prise avant la détection (ne contenant pas d'objets) [9]. Une binarisation de seuil σ permet alors de faire ressortir les objets de l'avant plan comme on peut le voir dans la figure 1.3. Suite à cette binarisation, il est possible de détecter des objets présents dans l'avant-plan et de les localiser facilement. Ce procédé est souvent utilisé dans le traitement de vidéos de scènes d'intérieur ainsi qu'en contrôle qualité par vision, car l'environnement y est contrôlé.

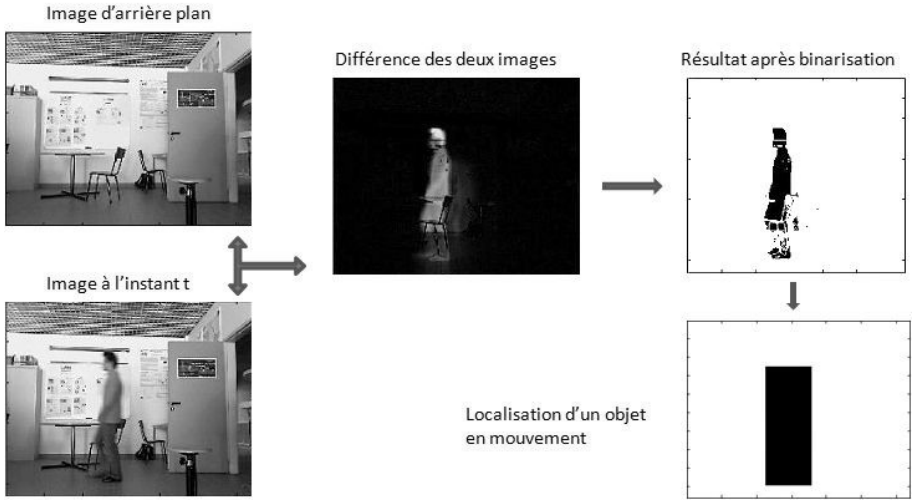


FIGURE 1.3 – Localisation par estimation de l'avant plan

Discussion

On peut voir que les algorithmes de localisation peuvent fonctionner sans avoir recours à des informations *a priori* sur les objets à localiser mais ils ne peuvent évidemment pas permettre dans ce cas de cibler la détection d'un seul type d'objet. Afin de pouvoir localiser certains objets d'intérêt, il faut avoir recours à une base de données permettant d'apprendre la nature de ces objets. Dans ce cas, on parle d'algorithme supervisé.

1.2.3 Les algorithmes de reconnaissance et de catégorisation

Les algorithmes de reconnaissance répondent au problème de la prédiction de la classe d'un ou plusieurs objets présents dans une image. La différence entre les

algorithmes de reconnaissance et de catégorisation est minime. Elle se situe au niveau de la précision souhaitée dans les résultats. Ainsi, un algorithme de reconnaissance fera la différence entre deux voitures de marques différentes alors qu'un algorithme de catégorisation les affectera dans la même catégorie. En pratique, la formalisation du problème est identique pour la reconnaissance et la catégorisation. Ces algorithmes ayant besoin d'une base de données pour fonctionner, ils sont par nature supervisés.

Formalisation du problème

Les algorithmes de reconnaissance doivent, à partir d'une image contenant un objet, lui assigner une classe $Y \in \mathcal{Y}$. L'ensemble $\mathcal{Y}$ est défini à partir de la base de données B et chacune des classes Y est apprise empiriquement à partir des exemples. On peut donc résumer cela ainsi :

$$Reconnaissance : B, I \mapsto Y$$

Ces algorithmes ne sont pas capables d'identifier plusieurs objets à la fois. Si l'on souhaite reconnaître plusieurs objets dans une même image, il va falloir les reconnaître les uns après les autres. Pour cela, il faut donner en entrée un ensemble de localisations (provenant d'une localisation non supervisée préalable) en plus de l'image elle-même. On a alors ceci :

$$Reconnaissance : B, I, Z_i \mapsto Y_i$$

Les algorithmes de reconnaissance doivent faire face à deux challenges majeurs. D'une part, ils doivent être capables de faire la différence entre les classes apprises. D'autre part, ils doivent être capables de différencier les objets appartenant à l'ensemble d'apprentissage et les objets inconnus. Les sorties possibles de ce type d'algorithme sont alors les suivantes :

- $Y \in \mathcal{Y}$, quand l'algorithme reconnaît une des classes apprises,
- une décision *Autre*, quand l'algorithme reconnaît un objet n'appartenant pas à l'ensemble d'apprentissage,
- une décision *Ambigu*, quand l'algorithme reconnaît un une des classes, mais n'arrive pas à décider laquelle.

$$\mathcal{Y}^+ = \mathcal{Y} \cup Autre \cup Ambigu$$

Afin d'avoir plus ou moins confiance dans les résultats fournis par l'algorithme de reconnaissance, un paramètre μ peut accompagner la réponse de l'algorithme. Ce paramètre est généralement compris dans l'intervalle $[0, 1]$, 1 représentant la

confiance maximale dans les résultats de l'algorithme. De même, un paramètre λ peut être mis en entrée de l'algorithme. Ce paramètre fonctionne alors comme un seuil et l'algorithme renvoie une réponse *Ambigu* si la confiance accordée au résultat est inférieure à λ. Finalement, on a :

$$Reconnaissance : B, I, Z_i \mapsto Y_i, \mu_i \tag{1.2}$$

ou

$$Reconnaissance : B, I, Z_i, \lambda \mapsto Y_i \tag{1.3}$$

L'algorithme peut également fournir un vecteur de résultats $\mu_{i_c} = \{\mu_c\}_{c \in \mathcal{Y}}$, avec μ_c la probabilité d'appartenance de l'objet i à la classe c. Cela laisse ainsi la possibilité de choisir la classe ultérieurement, en prenant par exemple la classe ayant la probabilité maximum.

$$Reconnaissance : B, I, Z_i \mapsto \mu_{i_c} \tag{1.4}$$

Exemple

Une méthode fréquemment utilisée en reconnaissance d'objets consiste à effectuer une reconnaissance sur des données locales dans l'image [10, 11], en prenant en compte leur répartition dans l'image. Pour cela, un détecteur de points d'intérêt est utilisé afin de localiser les zones de l'image qui vont être décrites. Suite à cette détection de points d'intérêt, les voisinages de ces points sont décrits au moyen d'un descripteur invariant. Un classifieur est ensuite utilisé sur les descripteurs dans le cadre des algorithmes de reconnaissance.

Afin de pouvoir effectuer une catégorisation, on peut également utiliser un « dictionnaire visuel » et une représentation en « patchs » [12, 5]. Pour cette méthode, on définit tout d'abord lors de l'apprentissage un dictionnaire de « mots visuels » obtenus par des méthodes de regroupement des vecteurs de descripteurs. On assigne ensuite chaque vecteur de l'image à un mot visuel, puis on étudie la répartition de ces mots visuels au moyen d'un histogramme indiquant le nombre de fois où les mots visuels apparaissent dans l'image. On peut voir le principe de l'utilisation de sacs de mots dans la figure 1.4. Enfin, un classifieur est utilisé sur la répartition des mots visuels afin d'obtenir la catégorisation.

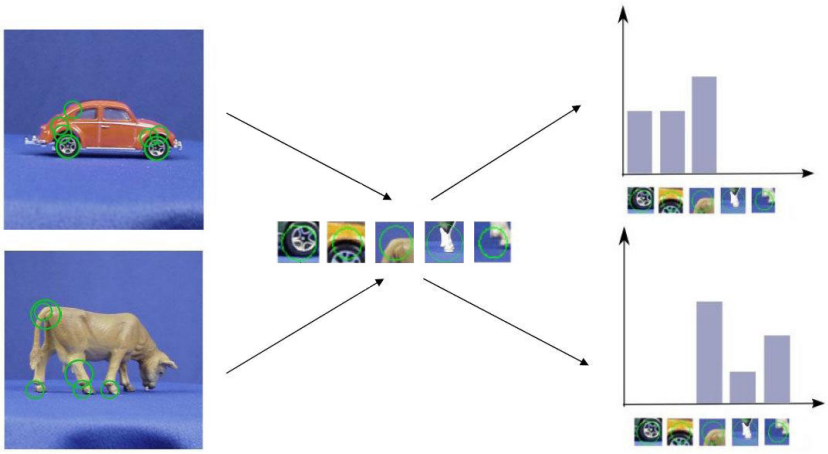


FIGURE 1.4 – Exemple de reconnaissance utilisant des patchs de l'objet [12] : chaque point d'intérêt est associé à un mot visuel, puis un histogramme indiquant la répartition des mots visuels dans l'images est calculé. La classe de l'objet est déduite de l'histogramme.

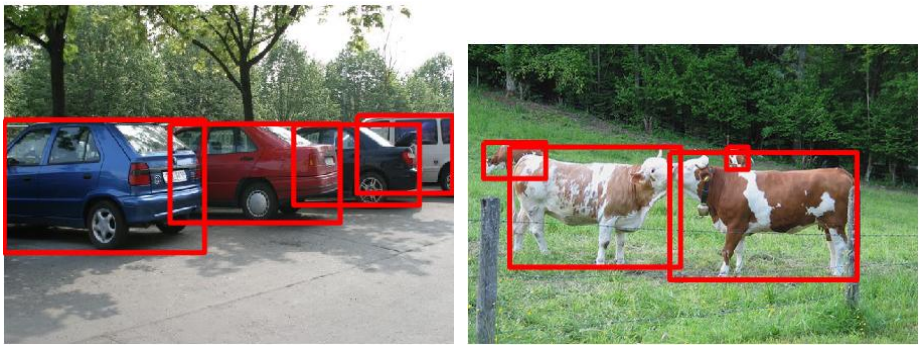
Discussion

Comme nous avons pu le voir, les algorithmes de reconnaissance et de catégorisation peuvent utiliser de nombreuses étapes : détection de points d'intérêt, descripteurs, apprentissage ... Ces algorithmes sont donc plus complexes que les algorithmes de localisation non supervisés vus précédemment. Il est alors indispensable de pouvoir évaluer les algorithmes de reconnaissance afin de vérifier que l'enchaînement de ces différents composants soit correct.

De plus, le problème des algorithmes de reconnaissance est que leur utilisation se limite à un seul objet à reconnaître dans l'image. Dans le cas où plusieurs objets sont présents dans l'image, il faut alors faire appel aux algorithmes de localisation afin d'isoler les différents objets du reste de la scène présentée dans l'image.

1.2.4 Les algorithmes supervisés de localisation

Comme nous l'avons vu auparavant, le but des algorithmes de localisation est de déterminer l'emplacement d'un ou de plusieurs objets dans une image. Les algorithmes supervisés se limitent généralement à la localisation d'un seul type d'objet et vont utiliser une base de données disposant d'images du type d'objet à localiser. On peut cependant chercher à localiser plusieurs objets de la même classe dans une seule image comme on peut le voir dans la figure 1.5.



(a) Localisation de véhicules (b) Localisation d'animaux

FIGURE 1.5 – Exemples de résultats de localisation supervisée

Formalisation du problème

Les algorithmes supervisés de localisation fournissent, à partir d'une image I, une liste de localisations d'objets $\{Z_i\}$. Les objets que l'on souhaite localiser sont appris au préalable au moyen d'une base d'apprentissage B contenant des exemples d'images avec les objets d'intérêt.

$$Localisation : B, I \mapsto \{Z_i\} \tag{1.5}$$

L'algorithme peut renvoyer la confiance accordée dans la localisation de l'objet au moyen d'un paramètre μ. La décision peut alors être effectuée en dehors de l'algorithme. L'algorithme peut également utiliser un critère λ en entrée, permettant un seuillage de la fonction de décision afin de savoir si un objet est présent. Le nombre d'objets localisés dans l'image I dépend de cette fonction de décision et est donc lié au seuil λ. On peut donc résumer cela ainsi :

$$Localisation : B, I \mapsto \{Z_i, \mu_i\} \tag{1.6}$$

ou

$$Localisation : B, I, \lambda \mapsto \{Z_i\} \tag{1.7}$$

Exemples

Un exemple d'algorithme supervisé de localisation d'objets dans une image est l'algorithme utilisé dans [13, 14, 15], dans lequel une fenêtre de taille fixe parcourt

l'ensemble de l'image mise à différentes échelles. Une classification est ensuite effectuée afin de dire si un objet de la classe apprise est présent ou non dans la fenêtre. Ainsi, dans [13, 14], la classification a lieu avec un réseau de neurones tandis que dans [15], on utilise un histogramme des gradients orientés, similaire au descripteur SIFT, combiné avec un séparateur à vaste marge (SVM). Toutes les fenêtres où l'on a trouvé un objet sont ensuite fusionnées afin d'obtenir une localisation de tous les objets dans l'image par des boîtes englobantes.

Une autre possibilité est de déterminer la localisation par reconnaissance de points d'intérêt comme le font Jurie *et coll.* dans [16]. Ainsi, des points d'intérêt sont calculés sur l'ensemble de l'image. Au voisinage de chaque point d'intérêt, on essaye de reconnaître un objet préalablement appris. Une boîte englobante entourant les points d'intérêt correctement reconnus est calculée. Nous pouvons voir en haut de la figure 1.6, l'image apprise de l'objet à localiser et son contour, et en bas les points d'intérêt extraits d'une image test et la localisation des objets qui en découle.

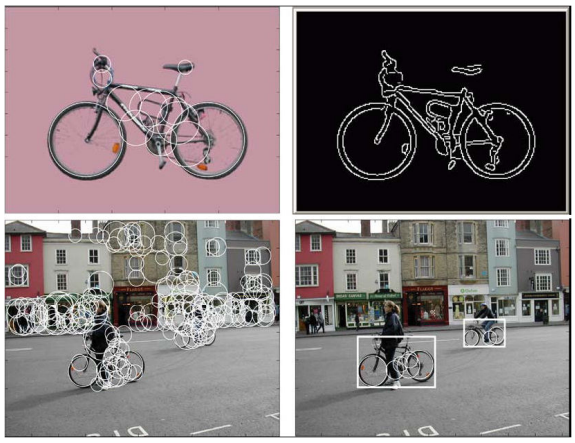


FIGURE 1.6 – Exemples de localisation par reconnaissance de points d'intérêt (image tirée de [16]).

Discussion

Nous avons vu que les algorithmes supervisés de localisation apportent plus d'informations que les algorithmes de reconnaissance. En effet, on y retrouve des techniques assez proches de celles utilisées en reconnaissance : l'utilisation de descripteurs et la classification en sont des étapes essentielles. Cependant, afin d'effectuer une localisation, d'autres étapes sont ajoutées comme le parcours d'une fenêtre dans

l'image ou l'étude de l'agencement spatial des points d'intérêt [16, 17, 18, 19].

Comme nous venons de le voir, les algorithmes supervisés de localisation sont des algorithmes complexes, utilisant différents composants.

1.2.5 Les algorithmes de localisation multi-classes

On qualifiera d'algorithme de localisation multi-classes un algorithme capable à la fois de localiser et de reconnaître un ou plusieurs objets de classes différentes présents dans une image.

Formalisation du problème

Le but d'un algorithme d'interprétation d'image est de fournir à la fois la localisation et la classe d'un ou plusieurs objets dans une image I. Cet algorithme est forcément supervisé et utilise donc une base de données B pour l'apprentissage des classes des objets à localiser dans l'image. On a alors :

$$Localisation\ multi-classes : B, I \mapsto \{Y_i, Z_i\}$$

L'algorithme peut également fournir un paramètre μ pour chaque objet localisé, indiquant la confiance que l'on a dans le résultat obtenu pour cet objet. On peut également fournir en entrée un paramètre λ qui fonctionnera comme un seuil sur la fonction de décision. Au final, on a donc ceci :

$$Localisation\ multi-classes : B, I \mapsto \{Y_i, Z_i, \mu_i\} \tag{1.8}$$

ou

$$Localisation\ multi-classes : B, I, \lambda \mapsto \{Y_i, Z_i\} \tag{1.9}$$

Exemple

Une méthode simple consiste à utiliser plusieurs fois un algorithme supervisé de localisation, avec une classe apprise différente à chaque fois. Il est également possible d'utiliser un classifieur qui va apprendre directement plusieurs classes. Ceci est réalisé, par exemple, par Torralba *et coll.* dans [20]. Un algorithme de boosting est utilisé afin de faire une localisation robuste. Pour pouvoir reconnaître et localiser plusieurs classes tout en réduisant le nombre de classifieurs nécessaires, un apprentissage est fait en partageant les données issues des différentes classes. Cela permet de faire de l'interprétation d'images avec différentes classes d'objets, mais également

avec différentes vues d'un même objet. On peut voir sur la figure 1.7 des images correctement interprétées.

FIGURE 1.7 – Exemple d'interprétation d'images (images tirées de [20])

Discussion

Les algorithmes de localisation multi-classes sont les algorithmes les plus complets. Ils sont capables de fournir à la fois la localisation et la classe de plusieurs objets dans une image. C'est à l'évaluation de tels algorithmes que nous nous sommes intéressés, et à l'évaluation d'algorithmes plus simples de localisation ou de reconnaissance qui en sont des briques de bases.

1.2.6 Les autres algorithmes d'interprétation d'images

Les algorithmes que nous avons présentés jusqu'à présent sont les algorithmes les plus usuels en interprétation d'images. Cependant, il en existe d'autres qui vont extraire de l'information sur les objets contenus dans une image. Ces algorithmes sont généralement dérivés des algorithmes présentés précédemment.

Détection

Nous appellerons algorithmes de détection, un algorithme dont le but est de déterminer la présence ou l'absence d'un objet dans une image. Ces algorithmes peuvent être supervisés ou non, multi-classes ou non.

Dans le cas non supervisé, l'algorithme de détection donne en résultat une réponse de type $Vrai$ ou $Faux$ indiquant si un objet est présent dans l'image.

$$Detection : I \mapsto (Vrai|Faux) \tag{1.10}$$

Dans le cas supervisé, l'algorithme de détection ne doit renvoyer $Vrai$ que si un objet d'une classe apprise, à partir des images de la base d'apprentissage B, est présent dans l'image. De plus, l'algorithme peut utiliser un paramètre μ ou λ, comme précédemment, afin d'indiquer le degré de confiance que l'on a dans le résultat. On peut donc résumer cela ainsi :

$$Detection : B, I \mapsto (Vrai|Faux), \mu \tag{1.11}$$

ou

$$Detection : B, I, \lambda \mapsto (Vrai|Faux|Ambigu)$$

Ce genre d'algorithme peut, par exemple, être utilisé afin de détecter la présence d'un individu afin de réguler l'éclairage ou le chauffage dans une pièce. Pour certains algorithmes, il n'est pas nécessaire de connaître la position de l'individu. Pour ce faire, on peut alors utiliser un algorithme de reconnaissance et renvoyer le résultat $Vrai$ uniquement si un individu est reconnu dans l'image.

Suivi

Les algorithmes de suivi sont des algorithmes dérivés des algorithmes de localisation. En effet, leur but va être de localiser un ou plusieurs objets dans une vidéo et de le suivre tout au long de son déplacement (voir figure 1.8). Le suivi est généralement plus robuste que la localisation. En effet, le fait de disposer d'une vidéo permet d'avoir plus d'informations qu'une image seule. Ainsi, on peut augmenter la confiance que l'on a dans un résultat de localisation s'il est proche du résultat sur les images précédentes.

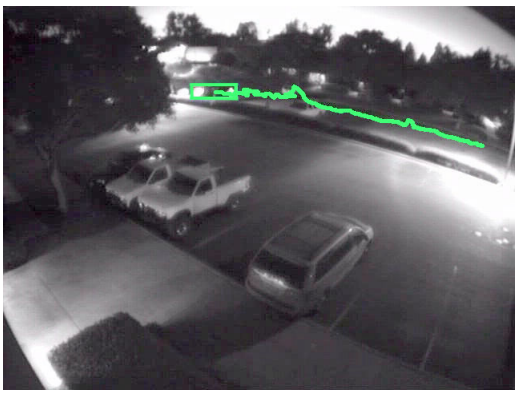


FIGURE 1.8 – Exemple de suivi d'un véhicule dans un parking

Le suivi d'objets permet entres autre d'étudier le comportement d'individus ou des trajectoires anormales d'objets. Par exemple, le suivi embarqué dans un véhicule

va permettre d'alerter le conducteur d'un comportement anormal d'un objet (un autre véhicule) dans son champ de vision.

1.3 Conclusions

Nous avons vu que l'interprétation regroupe un grand nombre de méthodes répondant chacune à une problématique différente. En effet, le terme même d'interprétation d'image est très mal défini par la communauté. Nous avons donc essayé de proposer une définition générale de l'interprétation. Nous avons également proposé une classification de plusieurs algorithmes d'interprétation d'images particuliers, notamment la localisation et la reconnaissance : les algorithmes de reconnaissance cherchent à savoir quels types d'objets sont présents dans une image, alors que les algorithmes de localisation cherchent à savoir où ils se situent.

Devant ce foisonnement, le problème de l'évaluation de l'interprétation devient primordial et soulève de nombreuses questions. Il faut en effet se demander dans quelle mesure nous sommes, par exemple, capables de mettre en évidence qu'une méthode de localisation ou de reconnaissance est meilleure qu'une autre.

Dans la suite de cette thèse, nous allons nous intéresser à l'évaluation des algorithmes de localisation multi-classes. Nous proposons de séparer le problème de l'évaluation de la localisation d'une part, et de l'évaluation de la reconnaissance d'autre part. Le chapitre suivant présente des éléments de l'état de l'art concernant l'évaluation de ces algorithmes.

Chapitre 2

État de l'art sur l'évaluation de l'interprétation d'images

Ce chapitre présente les différentes méthodes rencontrées dans l'état de l'art pour l'évaluation d'algorithmes d'interprétation d'images. Nous nous sommes intéressés à l'évaluation locale réalisée sur un seul résultat d'interprétation, et aussi à l'évaluation globale réalisée à partir d'une base de données.

Sommaire

2.1 Introduction

Nous avons vu que le terme *interprétation* d'images regroupe plusieurs types d'algorithmes. Nous nous intéresserons successivement dans ce chapitre à l'évaluation des algorithmes de localisation, de reconnaissance et enfin de ceux effectuant

les deux en même temps.

Deux types d'évaluation peuvent être distingués dans la littérature : l'évaluation d'un résultat d'interprétation ou bien une évaluation plus globale portant sur plusieurs résultats d'interprétation. L'évaluation d'un résultat d'interprétation en mode supervisé consiste à quantifier la similarité entre un résultat d'interprétation donné et une vérité terrain. C'est ce type d'évaluation qui est le plus fréquemment rencontré lorsqu'on s'intéresse aux algorithmes de localisation. L'évaluation portant sur un grand ensemble d'images est quant à elle plus souvent rencontrée lorsqu'on s'intéresse à la reconnaissance. L'algorithme fournit plusieurs résultats d'interprétation pour chaque image de la base de données, et l'évaluation va consister à obtenir un score global à partir des évaluations de chaque résultat d'interprétation.

Avant de détailler les différentes méthodes d'évaluation d'algorithmes d'interprétation pouvant être trouvées dans la littérature, nous présentons dans le paragraphe suivant les difficultés spécifiques liées à cette évaluation. Ensuite, les différentes méthodes d'évaluation existantes sont présentées : les méthodes permettant d'évaluer un résultat d'interprétation puis les méthodes permettant d'évaluer un algorithme dans son ensemble. Enfin, une conclusion met en évidence les meilleures approches *a priori*.

2.2 Les difficultés de l'évaluation d'algorithmes d'interprétation

Quel protocole doit-on mettre en place pour pouvoir évaluer différents algorithmes d'interprétation ? Quels sont les outils nécessaires à cette évaluation ? Comment comparer des algorithmes fournissant des résultats de formes différentes comme c'est le cas pour la localisation par exemple ?... Autant de questions auxquelles il convient de réfléchir.

Nous allons voir dans cette partie les différents problèmes que pose l'évaluation d'un résultat de localisation, et les moyens d'évaluer simultanément des algorithmes utilisant des types différents de représentation de la localisation.

2.2.1 Méthodologie

La méthodologie la plus complète afin d'effectuer l'évaluation d'un algorithme consiste à effectuer une évaluation par diagnostic [21]. Cette méthode consiste à faire

un très grand nombre de mesures quantitatives sur la qualité des résultats fournis par les algorithmes d'interprétation. Nous nous intéresserons plus spécifiquement aux métriques d'évaluation supervisée qui permettent de comparer le résultat d'un algorithme d'interprétation à la vérité terrain. Pour réaliser cette tâche, il faut réunir trois éléments :

- Une base d'images test,
- Une base de vérités terrain correspondantes,
- Une métrique d'évaluation.

La constitution de la base d'images et des vérités terrain correspondantes nécessite beaucoup de ressources, notamment d'un point de vue organisation, et de précision afin d'être pertinente comme nous allons le voir par la suite.

Cette méthode d'évaluation par diagnostic permet de juger différents algorithmes dans des conditions similaires, et donc d'avoir des résultats fiables pour les comparer. Cependant, il faut s'assurer que les images dans la base de données soient représentatives et en nombre suffisant, que la vérité terrain soit suffisamment précise et que la métrique d'évaluation soit pertinente dans toutes les situations (type d'image, type d'algorithme, robustesse à différents artefacts...) sans favoriser une méthode. Le choix de la métrique d'évaluation est très important puisqu'elle conditionne intégralement le jugement réalisé sur les algorithmes.

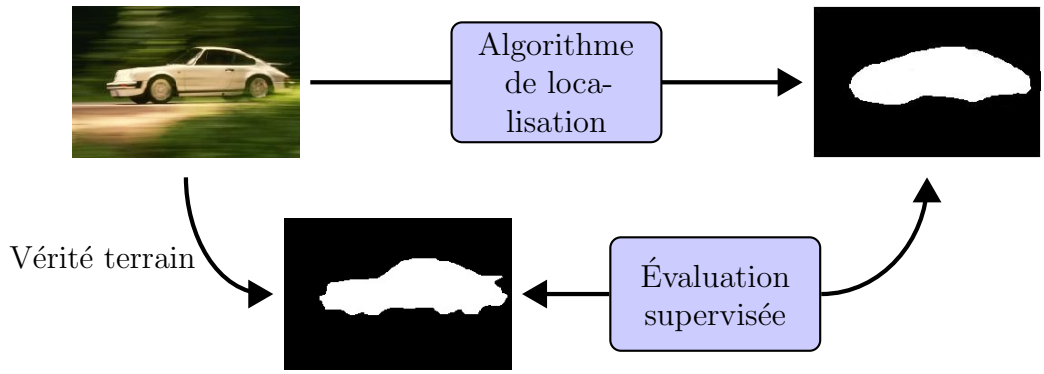


FIGURE 2.1 – Principe de l'évaluation de la localisation par diagnostic

2.2.2 Base d'images

La constitution de bases d'images pertinentes est essentielle pour espérer une bonne évaluation. Différentes difficultés sont fréquemment rencontrées.

Réalisation des bases d'images

La réalisation de bases d'images a pour but de rendre compte des différents cas d'utilisation des algorithmes d'interprétation. Pour être pertinente, la base doit contenir beaucoup d'images illustrant les cas simples comme les cas les plus compliqués (présence de bruit, occultation, forte variation de luminosité...). La création de bases d'images peut alors se faire de deux façons différentes : en utilisant des images synthétiques ou des images réelles.

Les bases contenant des images réelles sont bien évidemment celles reflétant le mieux le contexte d'utilisation des algorithmes. L'évaluation d'algorithmes doit donc se faire, autant que possible sur ces bases d'images. Cependant, la réalisation de telles bases est compliquée. En effet, il faut d'abord réunir un grand nombre d'images représentatives de tous les cas d'utilisation des algorithmes. Il faut ensuite annoter chaque image de la base de données avec le plus d'informations possible : catégorie, classe, localisation, nombre d'objets... Cette tâche doit être faite manuellement pour chaque image. Des compétitions telles que Pascal VOC Challenge [1], Robin [2], PETS [3], ETISEO [4] ou bien TrecVideo [5], qui ont pour objectif de comparer les performances d'algorithmes d'interprétation de l'état de l'art, ont mis en place ce genre de bases de données.

La création de bases d'images synthétiques pose moins de problèmes. Néanmoins, ces bases sont évidemment moins représentatives des cas réels d'utilisation des algorithmes. Pour créer une base d'images synthétiques, on référence tout d'abord les différents scénarios et altérations que l'on souhaite étudier. On peut ensuite créer les images et les vérités terrain correspondant à chaque scénario avec ou sans altération. L'avantage est que nous disposons de vérités terrain fiables et que l'on maîtrise l'importance des différentes altérations imposées. Cela permet d'étudier, par exemple, la robustesse d'un algorithme face à un certain type d'altération.

Confiance accordée aux vérités terrains

Si les vérités terrain sont parfaites et uniques dans le cas des bases d'images synthétiques, ce n'est pas le cas des vérités terrain générées par des humains pour des bases d'images réelles. On peut en effet se poser la question de la confiance accordée

1. http://www.pascal-network.org/challenges/VOC/databases.html
2. http://robin.inrialpes.fr/datasets.php
3. http://www.cvg.cs.rdg.ac.uk/datasets/index.html
4. http://www-sop.inria.fr/orion/ETISEO/download.htm
5. http://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html

aux vérités terrain, en particulier au niveau de la précision de la localisation des objets dans une image. Les appréciations peuvent différer entre deux personnes et donc les vérités terrain également. Par exemple, dans le contour d'une voiture, un expert pourra inclure l'aileron du véhicule dans la vérité terrain tandis qu'un autre ne l'inclura pas. La figure 2.2 illustre ce phénomène.

FIGURE 2.2 – Différence de localisation d'un même objet par différents experts

2.2.3 Représentation de la localisation

La localisation d'un objet, c'est-à-dire la forme que peut prendre l'espace $\mathcal{Z}$, peut se faire de quatre façons différentes :

- Centre de l'objet,
- Boîte englobante,
- Contour,
- Masque.

Parmi ces différentes représentation possibles, la plus utilisée est celle exploitant des boîtes englobantes. En effet, celle-ci est la plus simple à mettre en place mais n'apporte pas autant de précision que le contour ou le masque. Cependant, suivant l'application visée, il arrive que les autres méthodes soient également utilisées.

Centre de l'objet

La caractérisation la moins précise consiste à localiser un objet par son centre (x_c, y_c), comme on peut le voir dans la figure 2.3. Cette méthode est peu utilisée. En effet, cette méthode est trop peu précise et la méthode de calcul du centre influe beaucoup sur le résultat obtenu. On peut, par exemple, dire que le point recherché est le centre d'une boîte englobante, ou bien le barycentre d'un masque, auquel cas, cette localisation fait suite à une autre méthode de localisation plus précise.

On peut aussi calculer le centre comme étant le point moyen d'un ensemble de points d'intérêt faisant partie de l'objet. Quoi qu'il en soit, cette méthode ne nous renseigne pas sur la géométrie de l'objet. Cette représentation a alors un intérêt limité.

$\Longleftrightarrow (70, 138)$

FIGURE 2.3 – Localisation par le centre de l'objet

Boîte englobante

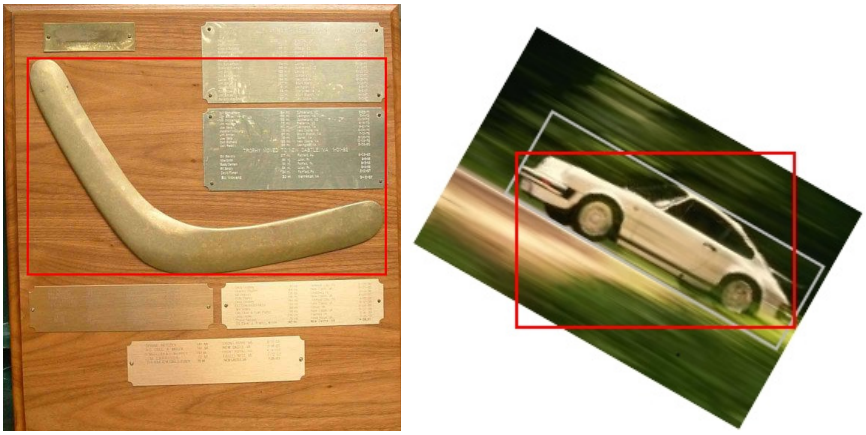
La caractérisation par des boîtes englobantes consiste à définir la position de l'objet par une boîte rectangulaire dont les côtés sont parallèles aux bords de l'image. Cette boîte est définie par un couple de coordonnées dans l'image $\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$. Cette boîte contient alors l'objet localisé (voir Fig. 2.4).

$\Longleftrightarrow \{(43, 37), (107, 242)\}$

FIGURE 2.4 – Localisation par une boîte englobante

Cette méthode est assez précise tout en étant facile à implémenter. En effet, elle nous renseigne sur la position et la géométrie générale de l'objet, c'est-à-dire sa hauteur et sa largeur. Elle pose cependant certains problèmes puisque l'objet peut ne pas remplir majoritairement la boîte englobante. Ce problème est particulièrement pénalisant avec les objets fortement non convexes. De plus, l'utilisation des boîtes englobantes n'est pas robuste aux rotations. On peut voir les problèmes liés à l'utilisation des boîtes englobantes dans la figure 2.5. Certains problèmes peuvent

aussi se poser lorsque la boîte contient également l'ombre portée de l'objet.



(a) Problème lié à la convexité de l'objet : une grande partie de la boîte englobante représente le fond plutôt que l'objet

(b) Problème lié à la rotation

FIGURE 2.5 – Problèmes liés à la localisation par une boîte englobante

Le fait d'utiliser une boîte englobante lors de la phase d'apprentissage peut au contraire, dans certains cas, permettre de faciliter la localisation d'un objet par la suite. En effet, la boîte englobante contient plus d'informations que l'objet en lui-même, avec par exemple le fond qui lui est généralement associé (route pour une voiture, pâturage pour une vache...). Cependant, cela peut aussi se révéler être un inconvénient puisqu'il est alors plus difficile de localiser ou reconnaître des objets s'ils ne sont pas dans leurs environnements habituels.

Contour

La caractérisation utilisant des contours est beaucoup plus précise que les boîtes englobantes. Le principe est de délimiter le contour extérieur de l'objet au moyen d'une frontière. Suivant les algorithmes, ce contour peut être plus ou moins complexe (voir Fig. 2.6 : polygones fermés convexes, polygones fermés non convexes, tracés fermés). On peut donc voir que cette méthode est plus précise puisqu'elle délimite de façon plus étroite la frontière entre l'objet et l'arrière plan.

(a) Polygones fermés convexes (b) Tracés fermés

FIGURE 2.6 – Localisation par un contour

Masque

Enfin, la localisation peut être représentée sous la forme d'un masque. Ce masque est une image binaire de la même taille que l'image originale. Chaque pixel de ce masque indique alors la présence ou l'absence de l'objet dans l'image. Cela revient à définir des régions contenant un objet et des régions n'en contenant pas comme on peut le voir dans la figure 2.7. Ce genre de localisation est le plus précis et autorise une représentation de la localisation de l'objet avec des parties occultées. Dans ce cas, il peut y avoir des trous dans le masque de l'objet.

FIGURE 2.7 – Localisation par un masque

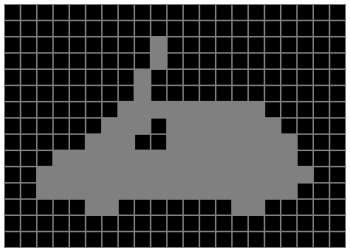
2.2.4 Équivalence des représentations de localisation

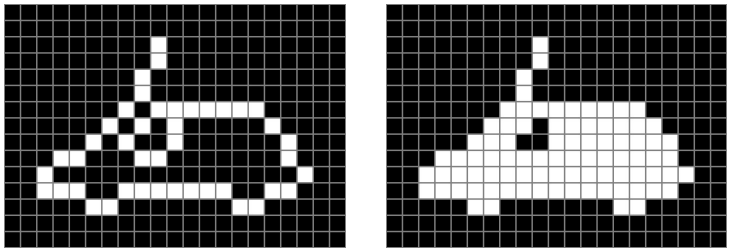
Nous avons vu qu'il existe plusieurs types de localisation d'un objet dans une image. Il est donc important de pouvoir passer d'un type de localisation à un autre afin de pouvoir utiliser des méthodes d'évaluation spécifiques (boîte englobante par exemple) pour évaluer tout algorithme de localisation, et ce indépendamment du type de représentation de la localisation obtenu en sortie de l'algorithme considéré. Dès lors,

il nous sera possible d'utiliser une seule métrique pour comparer plusieurs algorithmes.

Cette uniformisation n'est pas systématiquement souhaitable. En effet, nous avons vu que certains types de localisation sont moins précis que d'autres. Le résultat d'évaluation d'un contour par une méthode d'évaluation dédiée aux boîtes englobantes ne sera pas aussi précis que si on l'évalue avec une méthode prévue pour les contours.

Cependant, il pourra être intéressant dans certains cas de connaître les correspondances pouvant exister entre les divers types de localisation. Pour cela, nous allons nous intéresser à l'image présentée sur la figure 2.8.





$\{(3,3),(13,19)\}$

FIGURE 2.8 – Image originale et localisations associées

Passage entre un contour et une boîte englobante

Nous pouvons facilement obtenir une localisation par boîte englobante si nous disposons d'une localisation en contour. Pour cela, il nous suffit de considérer les coordonnées des pixels frontière, puis d'en extraire le couple $\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$. Ces coordonnées seront alors la délimitation de la boîte englobante (voir Fig. 2.9).

Nous pouvons également faire la démarche inverse, c'est-à-dire définir un contour à partir du couple $\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$. Bien évidemment, ce contour aura la

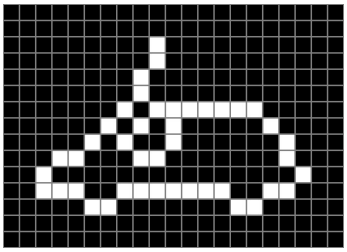

$\Rightarrow : \{(3,3),(13,19)\}$

FIGURE 2.9 – Passage d'une localisation contour à une boîte englobante

forme d'une boîte rectangulaire qui englobera l'objet à localiser. Ceci ne permettra donc pas de retrouver la précision offerte par les méthodes de localisation en contour (voir Fig. 2.10).

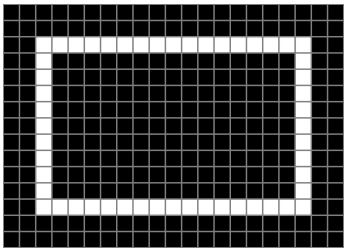
$\{(3,3),(13,19)\} \Rightarrow$ 

FIGURE 2.10 – Passage d'une boîte englobante à un contour

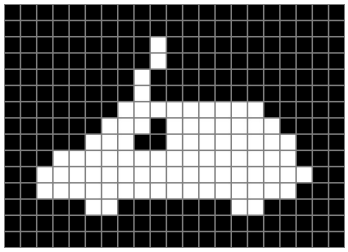
Passage entre un masque et une boîte englobante

Tout comme pour l'approche contour, il est facile d'obtenir une localisation par boîte englobante si nous disposons d'une localisation par masque. Pour cela, il nous suffit de considérer les coordonnées des pixels appartenant à l'objet et d'en extraire le couple $\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$. Ces coordonnées seront alors la délimitation de la boîte englobante (voir Fig. 2.11).

Comme précédemment, nous pouvons obtenir à partir d'un couple de coordonnées $\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$ un masque qui représentera cette fois encore une boîte rectangulaire (voir Fig. 2.12). Cette approche ne permettra pas non plus de gagner la précision offerte par les méthodes de localisation renvoyant un masque de localisation.

Passage entre un contour et un masque

Il est également possible de passer d'un contour à un masque lorsque les contours obtenus sont fermés (voir Fig. 2.13). L'intérêt réside ici dans le fait qu'on ne perd



$\Rightarrow \{(3,3),(13,19)\}$

FIGURE 2.11 – Passage d'un masque à une boîte englobante

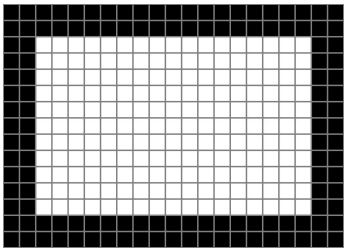
$\{(3,3),(13,19)\} \Rightarrow$ 

FIGURE 2.12 – Passage d'une boîte englobante à un masque

pas d'information lors du passage d'un type de localisation à l'autre.

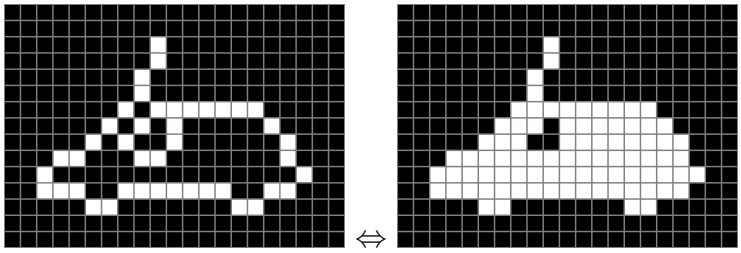


FIGURE 2.13 – Dualité entre un contour et un masque

Le passage inverse peut cependant poser problèmes. En effet, l'approche contour peut être interprétée de plusieurs façons différentes. La frontière peut être incluse dans l'objet ou être à l'extérieur de l'objet. De plus, on peut considérer que la frontière doit être 4-connexes ou bien 8-connexes (voir Fig. 2.14). Il convient donc de bien définir quel type de frontière on souhaite avoir avant de choisir une méthode de conversion entre un masque et un contour.

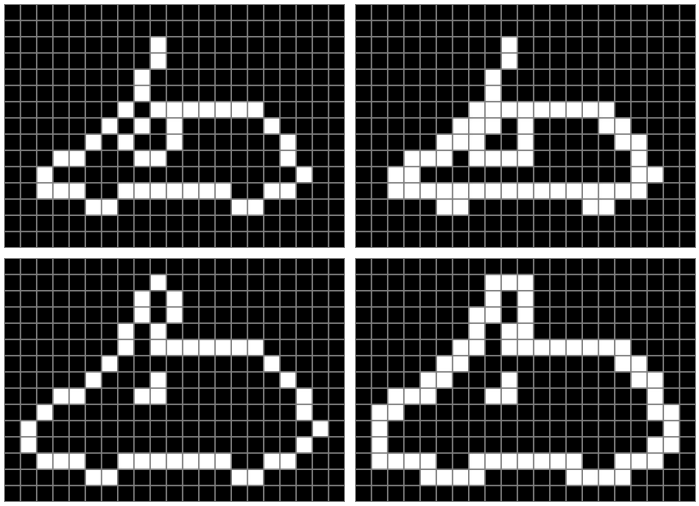


FIGURE 2.14 – Problèmes liés à la définition du contour : le contour peut être 4- ou bien 8-connexes, appartenir à l'objet ou bien au fond

2.2.5 Représentation de la reconnaissance

Un résultat de reconnaissance prend ses valeurs dans l'espace $\mathcal{Y}$, défini à partir de la base de données utilisée par l'application. Généralement, les éléments de l'espace $\mathcal{Y}$ prennent la forme d'un identifiant numérique ou bien d'une chaîne de caractères représentant l'objet tel que « avion », « personne » ou bien « bouteille » pour la base de données Pascal VOC challenge 2008 [22] par exemple.

Nous nous intéressons dans cette partie à trois représentations d'objets utilisées en interprétation d'images, et plus particulièrement en reconnaissance : les ensembles de points, les sacs de mots et les graphes.

Détecteurs de points d'intérêt

Afin de reconnaître deux objets, la comparaison systématique de tous les pixels entre deux images n'est pas envisageable pour des raisons évidentes d'augmentation rédhibitoire de temps de calcul. La reconnaissance peut être simplifiée en réduisant le domaine de recherche aux zones de l'image localement intéressantes. Les points référençant ces zones sont dénommés points d'intérêt (voir figure 2.15). Ces points d'intérêt correspondent à une rupture d'une ou plusieurs caractéristiques de l'image, tels que le niveau de gris ou de couleurs, la texture ou la géométrie [23].

FIGURE 2.15 – Exemples de détection de points d'intérêt

Harris

L'un des détecteurs de points d'intérêt les plus largement utilisé est le détecteur de Harris et Stephen [24], aussi connu sous le nom d'opérateur de Plessey. Harris et Stephen définissent une matrice qui sera largement reprise dans les détecteurs de points d'intérêt développés par la suite :

$$M^{HS} = G_\sigma * \begin{bmatrix} \frac{\partial I}{\partial x}^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \frac{\partial I}{\partial y}^2 \end{bmatrix} \tag{2.1}$$

avec I l'image , G_σ une fonction gaussienne :

$$G_\sigma(x, y) = e^{-\frac{x^2+y^2}{\sigma^2}} \tag{2.2}$$

où σ est l'écart type de la fonction gaussienne.

A partir de cette matrice, Harris et Stephen définissent un critère de sélection :

$$R^{HS} = Det(M^{HS}) - k \cdot Tr(M^{HS})^2 \tag{2.3}$$

avec $Det(\cdot)$ le déterminant, $Tr(\cdot)$ la trace et k un paramètre du critère. Plus la valeur de ce critère est haute et plus il est probable que le point considéré soit un point d'intérêt. Un seuillage sur la valeur de ce critère permet alors de sélectionner les points d'intérêts. Plus la valeur du seuil est basse et plus le nombre de points d'intérêts est grand.

Différence de Gaussiennes

Une autre méthode largement utilisée pour détecter des points d'intérêt est celle présente dans l'algorithme SIFT. Cette détection de points d'intérêt se fait dans l'espace des échelles [25]. Les points d'intérêt correspondent dans ce cas à des extrema dans la différence de gaussienne :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.4}$$

où k est une constante permettant de prendre en compte la différence d'échelle et $L(\cdot)$ est défini ainsi :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.5}$$

avec $*$ le produit de convolution, et G la fonction gaussienne suivante :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.6}$$

Cette première étape permet de récupérer un grand nombre de points d'intérêt. C'est pourquoi une deuxième étape permet de sélectionner les points d'intérêt les plus intéressants. Afin d'être sélectionné, un point doit satisfaire à deux conditions : la première concerne le contraste du point d'intérêt tandis que la seconde concerne le rayon de courbure du point d'intérêt. Pour la première condition, il faut que la valeur de $D(x, y, \sigma)$ soit supérieure à une valeur seuil (fixée à 0,03 par Lowe). Pour la seconde condition, on s'intéresse à la matrice hessienne de D, tout comme pour le critère de Harris et Stephen :

$$M^L = \begin{bmatrix} \frac{\partial D}{\partial x}^2 & \frac{\partial D}{\partial x}\frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial x}\frac{\partial D}{\partial y} & \frac{\partial D}{\partial y}^2 \end{bmatrix} \tag{2.7}$$

Le critère utilisé pour sélectionner les points, dépendant d'un paramètre r, est alors le suivant :

$$R^L = \frac{Tr(M^{HS})}{Det(M^L)} < \frac{(r+1)^2}{r} \tag{2.8}$$

Une valeur élevé du paramètre r permet de s'assurer que le point considéré est un point d'intérêt.

Descripteurs invariants

Le problème fondamental de la reconnaissance d'objets est de déterminer dans quelle mesure deux objets sont similaires, indépendamment de leurs positions, échelles

ou orientations dans l'image. Il en découle que les descripteurs doivent être invariants par rapport aux transformations géométriques (translations, rotations, homothéties). L'utilisation des descripteurs présentés dans cette partie permet de mesurer la similarité entre deux objets dans de telles conditions. Ces descripteurs peuvent être calculés sur l'ensemble de l'image. Cependant, ils sont plus souvent calculés au voisinage des points d'intérêt.

Les filtres complexes

Les filtres complexes [26] sont dérivés de l'équation suivante :

$$K_{mn}(x, y) = (x + iy)^m (x - iy)^n G(x, y) \tag{2.9}$$

avec $G(x, y)$ une fonction Gaussienne, x et y les coordonnées du pixel dans le filtre. Une quinzaine de filtres est utilisée avec $m - n < 6$, pour calculer le descripteur sur un voisinage de points d'intérêt. Une partie des filtres obtenus est présentée à la figure 2.16. Le vecteur descripteur final est composé de 15 éléments.

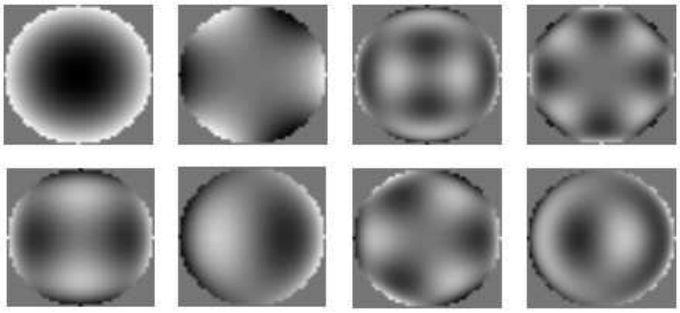


FIGURE 2.16 – Filtres complexes [26]

Filtres « Steerable » et invariants différentiels

Les descripteurs des filtres « Steerable » [27] et des invariants différentiels [28] sont calculés à partir de filtres de bases dérivés d'une fonction gaussienne. Chaque filtre de base, que l'on peut voir à la figure 2.17, est ensuite assigné d'un angle θ pouvant prendre les valeurs 0°, 60° et 120°.

Les filtres « steerable » sont multipliés par une carte de gain puis sommés pour obtenir le descripteur (voir la figure 2.18). Les filtres de bases sont calculés jusqu'au 4^e ordre pour obtenir 14 filtres de bases. On obtient 14 filtres, et non 16 comme on pourrait s'y attendre, car 2 des filtres sont invariants par rotation de 90°. Le vecteur descripteur final comporte donc 14 éléments.

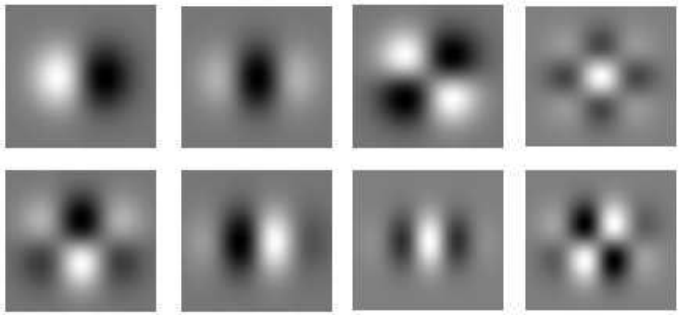


FIGURE 2.17 – 8 filtres de bases, les filtres manquants sont obtenus par rotation de 90°[27]

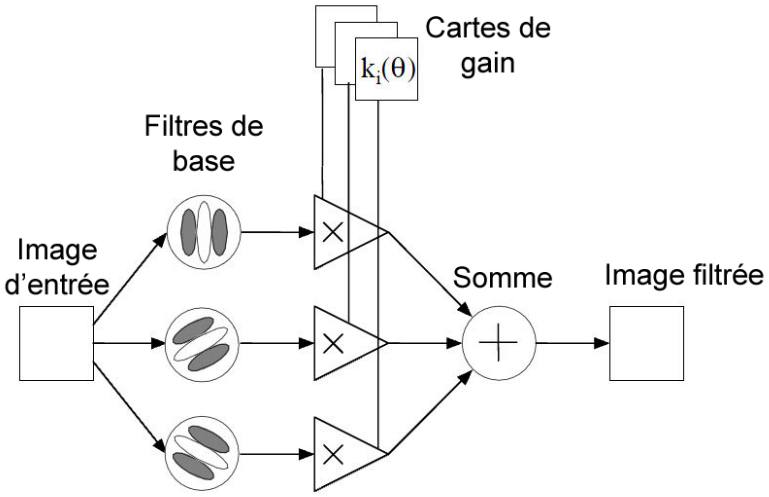


FIGURE 2.18 – Principe des filtres « Steerable » [27]

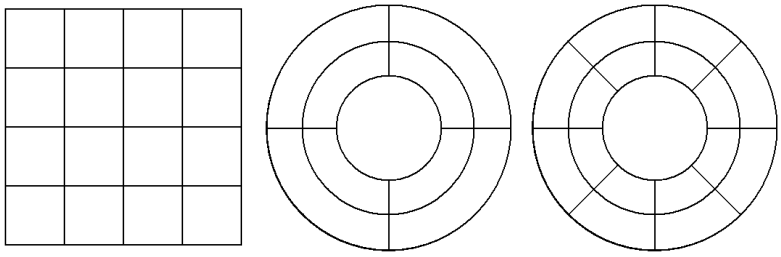
Les invariants différentiels reprennent le même principe, mais ne sont calculés que jusqu'au 3e ordre. Le vecteur descripteur final est alors composé de 12 éléments.

SIFT

SIFT [29, 25] est un descripteur largement utilisé et étudié depuis que Lowe l'a développé en 1999. SIFT est à la base constitué de quatre étapes : les trois premières correspondent à la détection et au choix des points d'intérêt tandis que la quatrième correspond à proprement parler au calcul du descripteur.

Le descripteur est calculé sur le voisinage de chaque point d'intérêt trouvé à l'issue des trois premières étapes. Pour cela, le voisinage est divisé par une grille 4*4 comme on peut le voir à la figure 2.19(a). Ensuite, le gradient est calculé sur

chacune des 16 localisations de la grille puis est quantifié selon un histogramme de 8 orientations. La concaténation de ces éléments nous permet d'obtenir un vecteur descripteurs de 128 éléments.



(a) SIFT (b) Shape context (c) GLOH

FIGURE 2.19 – Grilles utilisées pour différents descripteurs

ACP-SIFT

Le descripteur ACP-SIFT [30] est une variation du descripteur SIFT. L'idée de ce descripteur est d'utiliser la détection de points d'intérêt introduite par le descripteur SIFT, mais de compléter la quatrième étape par une analyse en composantes principales (ACP). Cette ACP permet de réduire la taille du vecteur descripteur (vecteur de 36 éléments), cela permet donc d'accélérer le cout de stockage et de calcul.

Contexte de forme

Le descripteur contexte de forme [31] fonctionne globalement comme le descripteur SIFT. La différence réside en trois points. Tout d'abord, la grille utilisée est différente : c'est une grille log-polaire avec 9 localisations, deux dans chacune des 4 directions plus une au centre comme on peut le voir à la figure 2.19(b). Ensuite, la quantification se fait selon 4 orientations au lieu de 8. Enfin, le descripteur est calculé non pas sur l'image initiale, mais une image présentant les contours extraits avec le détecteur de Canny [32]. Le vecteur descripteur obtenu comporte alors 36 éléments.

Histogramme des orientations et des gradients

Le descripteur « gradient location and orientation histogram » [4] est également une variante du descripteur SIFT. Il a été conçu dans le but d'améliorer la robustesse et la caractérisation du descripteur par rapport au descripteur SIFT. Il utilise, comme le descripteur contexte de forme, une grille log-polaire, mais avec plus de localisations : deux dans chacune des 8 directions plus une au centre, soit 17 localisations (voir la figure 2.19(c)). De plus, la quantification se fait selon 16 directions, ce qui amène

à un vecteur de 272 éléments. Cependant, une analyse en composantes principales ayant pour but de réduire la dimension de ce vecteur permet d'obtenir au final 128 éléments.

Représentations

Ensembles de points

Nous avons vu qu'un objet peut être décrit par un ensemble de points d'intérêts. Chacun de ces points d'intérêt est ensuite caractérisé grâce à un descripteur invariant.

La représentation d'un objet est alors, dans ce cas, l'ensemble des vecteurs caractéristiques des points d'intérêt.

Sacs de mots

La représentation en sac de mots [12, 5] consiste en le calcul d'un histogramme représentatif de l'objet (déjà abordé page 12). Pour cela, on commence par calculer des mots visuels qui vont constituer un dictionnaire, également appelé vocabulaire, lors d'une phase d'apprentissage. Ces mots visuels correspondent à des parties d'objets que l'on retrouve fréquemment, et sont généralement définis au moyen de méthodes d'agglomération (*clustering*).

Une fois ce dictionnaire visuel calculé, on associe alors chaque point d'intérêt d'un objet à un mot visuel. Il suffit alors de compter le nombre de points d'intérêt associés à chacun des mots visuels. On obtient finalement un histogramme, appelé sac de mots, qui peut être utilisé en tant que vecteur caractéristique de l'objet.

Graphes

Enfin, les graphes peuvent être utilisés afin de représenter un objet. Un graphe consiste à relier des points de l'objet par des arêtes. Ces points sont alors appelés les nœuds, ou sommets, du graphe. Le graphe peut être construit de différentes façons : une squelettisation d'une part, ou bien en reliant des points d'intérêts précédemment extraits d'autre part. Si l'on décide d'utiliser des points d'intérêts, alors, la manière dont sont reliés ces points est généralement paramétrable.

Le graphe obtenu est noté $G = (V, E)$, où V désigne l'ensemble des nœuds du graphe, tandis que E désigne l'ensemble des arêtes. Le graphe peut également être

labélisé, c'est-à-dire que chaque nœud et chaque arête du graphe peut avoir un label. On notera alors le graphe $G = (V, E, \mu, \nu, L_v, L_e)$, avec μ et ν des fonctions de label :

$$\begin{cases} \mu : V \to L_v & \text{fonction de label sur les nœuds} \\ \nu : E \to L_e & \text{fonction de label sur les arêtes} \end{cases} \tag{2.10}$$

La fonction μ permet d'assigner le vecteur caractéristique du point d'intérêt au noeud associé, tandis que la fonction ν permet d'assigner la distance entre deux points à l'arête qui les sépare. L'intérêt est alors que l'on dispose d'une information spatiale concernant les points en plus de leur caractérisation : un point d'intérêt sera proche d'autres points d'intérêt s'ils sont reliés par le graphe.

La matrice d'adjacence est souvent utilisée pour représenter le graphe. Cette matrice $M = (m_{i,j})$ est définie ainsi :

$$m_{i,j} = \begin{cases} \mu(v_i) & \text{si } i = j \\ \nu(v_i, v_j) & \text{si } i \neq j \text{ et } (v_i, v_j) \in E \\ 0 & \text{sinon} \end{cases} \tag{2.11}$$

2.3 Métriques d'évaluation d'un résultat de localisation

Nous avons vu qu'il existe différents types de localisation :

- Centre de l'objet : le centre de l'objet se représente par un point, c'est-à-dire un couple de coordonnées $P_c = (x_c, y_c)$,
- Boîte englobante : les boîtes englobantes sont généralement représentées par les deux points $\{P_{min}, P_{max}\} = \{(x_{min}, y_{min}), (x_{max}, y_{max})\}$,
- Contour : les contours sont généralement représentés par une image contenant la frontière entre l'objet et le fond. Les pixels de ces images prennent leurs valeurs dans $\{0, 1\}$. Les 0 représentent l'objet et le fond de l'image qui sont délimités par les pixels frontière de valeur 1,
- Masque : les masques sont des images binaires de type région. Les pixels prennent également leurs valeurs dans $\{0, 1\}$, les pixels 0 représentent le fond tandis que les pixels 1 représentent l'objet.

Ces différents types de localisation vont nécessiter l'usage de métriques d'évaluation différentes. Nous verrons dans cette partie les métriques existantes en fonction du type de localisation dans le cas de l'évaluation supervisée.

Si l'évaluation d'une localisation réalisée grâce au centre de l'objet peut se faire simplement par utilisation d'une distance dans le plan, l'évaluation d'une localisation par une boîte englobante, un contour ou un masque, est plus compliquée. Il existe quelques métriques spécifiques à l'évaluation de boîtes englobantes issues de [2]. Concernant les métriques pour les contours et les masques, un certain nombre est issu des méthodes d'évaluation de la segmentation [33]. Suivant qu'il s'agisse de segmentation en frontière ou bien de segmentation en région, les métriques peuvent être appliquées à de l'évaluation de la localisation par contour ou par masque.

2.3.1 Évaluation de la localisation par le centre de l'objet

Lorsqu'un objet est localisé par son centre, l'évaluation peut se faire avec une p-distance dans le plan :

$$D_p(P_l, P_{vt}) = \sqrt[p]{|x_l - x_{vt}|^p + |y_l - y_{vt}|^p} \tag{2.12}$$

où P_l est le centre de l'objet localisé de coordonnées $\{x_l, y_l\}$. De même, le point P_{vt} correspond au centre de l'objet dans la vérité terrain. Les cas particuliers $p = 1$ et $p = 2$ correspondent respectivement à la distance de Manhattan et à la distance euclidienne. Dans le cas particulier où $p = \infty$, on a alors $D_\infty(P_l, P_{vt}) = \max(|x_l - x_{vt}|, |y_l - y_{vt}|)$.

2.3.2 Évaluation de la localisation par une boîte englobante

Le projet ROBIN [2] définit trois métriques permettant de décider si une localisation est correcte au vue de la vérité terrain. Ces trois métriques correspondent à trois problèmes différents : la localisation, la complétude et la correction de la localisation. Les valeurs de ces métriques sont d'autant plus basses que la localisation est bonne, 0 étant un résultat optimal. Ils sont définis ainsi :

$$ROB_{loc}(BB_l, BB_{vt}) = \frac{2}{\pi} arctan(max(\frac{|x_l - x_{vt}|}{w_{vt}}, \frac{|y_l - y_{vt}|}{h_{vt}})) \tag{2.13}$$

$$ROB_{com}(BB_l, BB_{vt}) = \frac{|\mathcal{A}_l - \mathcal{A}_{vt}|}{max(\mathcal{A}_l, \mathcal{A}_{vt})} \tag{2.14}$$

$$ROB_{cor}(BB_l, BB_{vt}) = \frac{2}{\pi} arctan(|\frac{h_l}{w_l} - \frac{h_{vt}}{w_{vt}}|) \tag{2.15}$$

où BB_l est la boîte englobante du résultat de localisation, x_l, y_l les coordonnées du centre de la boîte englobante, w_l, h_l la largeur et la hauteur de la boîte englobante et $\mathcal{A}_l$ son aire. Les valeurs indicées par vt représentent leurs équivalents pour la vérité

terrain.

La métrique ROB_{loc} permet d'évaluer la distance entre le centre de la boîte englobante localisé par l'algorithme et celui de la vérité terrain. La métrique vaut 0 lorsque $x_l = x_{vt}$ et $y_l = y_{vt}$, c'est-à-dire lorsque le centre de la localisation et le centre de la vérité terrain sont confondus. La métrique ROB_{com} permet d'évaluer si les surfaces couvertes par les boîtes englobantes sont identiques puisque ROB_{com} vaut 0 lorsque $\mathcal{A}_l = \mathcal{A}_{vt}$. ROB_{cor} permet quant à lui d'évaluer la forme de la boîte englobante. En effet, la métrique vaut 0 lorsque $\frac{h_l}{w_l} = \frac{h_{vt}}{w_{vt}}$, c'est-à-dire lorsque les rapports hauteur/largeur des deux boîtes sont égaux. Ainsi, les trois métriques utilisées pour la compétition Robin sont complémentaires et permettent d'évaluer différentes altérations que peut subir une boîte englobante.

2.3.3 Évaluation de la localisation par un contour

Une partie des algorithmes d'évaluation de la segmentation s'intéresse à la segmentation en frontière. Cette segmentation donne des résultats proches des résultats obtenus avec une localisation en contour. La différence vient du fait que la localisation travaille exclusivement avec des contours fermés, ce qui n'est pas nécessairement le cas de la segmentation. Les évaluations correspondantes pourront donc être transposées au cas de l'évaluation de la localisation.

Métriques statistiques

Des métriques de type statistique ont été proposées pour permettre de quantifier une divergence entre deux images en prenant en compte le nombre et la valeur des pixels dans deux images. La première approche en évaluation de segmentation consiste à calculer l'écart entre le nombre de pixels des contours obtenus et le nombre de pixels des contours de la vérité terrain. Les trois mesures d'erreur les plus classiques, entre deux images I_l et I_{vt}, de support commun I, sont les suivantes [33, 34] :

– erreur de sur-détection

$$ErrSur(I_{vt}, I_l) = \frac{Card(I^{Fr}_{l\setminus vt})}{Card(I) - Card(I^{Fr}_{vt})} \tag{2.16}$$

– erreur de sous-détection

$$ErrSous(I_{vt}, I_l) = \frac{Card(I^{Fr}_{vt\setminus l})}{Card(I^{Fr}_{vt})} \tag{2.17}$$

– erreur de localisation

$$ErrLoc(I_{vt}, I_l) = \frac{Card((I^{Fr}_{vt\setminus l}) \cup (I^{Fr}_{l\setminus vt}))}{Card(I)} \tag{2.18}$$

où I_{vt}^{Fr} correspond à l'ensemble des pixels appartenant à la frontière dans l'image vérité terrain I_{vt}. L'ensemble des pixels appartenant à la frontière dans I_{vt} mais pas à la frontière dans I_l est noté $I_{vt\backslash l}^{Fr}$. Un bon algorithme doit minimiser les trois mesures d'erreurs simultanément.

La seconde approche consiste à calculer des mesures de divergence tels que le rapport signal bruit (SNR) ou bien l'erreur quadratique moyenne (RMS) [35, 36], soit en notant I le support commun aux deux images I_{vt} et I_l :

$$SNR(I_{vt}, I_l) = \left[\frac{1}{Card(I)} \sum_{k \in I} \frac{g_{I_l}(k)^2}{(g_{I_{vt}}(k) - g_{I_l}(k))^2}\right]^{\frac{1}{2}} \tag{2.19}$$

$$RMS(I_{vt}, I_l) = \left[\frac{1}{Card(I)} \sum_{k \in I} (g_{I_{vt}}(k) - g_{I_l}(k))^2\right]^{\frac{1}{2}} \tag{2.20}$$

où $g_{I_{vt}}(k)$ représente le niveau de gris du pixel k dans l'image I_{vt}. La métrique RMS peut s'étendre au calcul des distances L_q avec $q \in \mathbb{N}^*$:

$$L_q(I_{vt}, I_l) = \left[\frac{1}{Card(I)} \sum_{k \in I} |g_{I_l}(k) - g_{I_{vt}}(k)|^q\right]^{\frac{1}{q}} \tag{2.21}$$

On peut enfin compléter ces distances avec celles de Küllback, de Bhattacharyya et de Jensen [37], qui repose sur l'entropie de Rényi [38].

– Distance de Küllback :

$$KUL(I_{vt}, I_l) = \frac{1}{Card(I)} \sum_{k \in I} \big(g_{I_l}(k) - g_{I_{vt}}(k)\big) * Log\Big(\frac{g_{I_l}(k)}{g_{I_{vt}}(k)}\Big) \tag{2.22}$$

– Distance de Bhattacharyya :

$$BHA(I_{vt}, I_l) = -Log\Big(\frac{1}{Card(I)} \sum_{k \in I} \sqrt{g_{I_{vt}}(k) * g_{I_l}(k)}\Big) \tag{2.23}$$

– Distance de Jensen :

$$JEN(I_{vt}, I_l) = J\Big(\frac{I_{vt} + I_l}{2}, I_{vt}\Big) \tag{2.24}$$

avec

$$J(I_1, I_2) = H_\alpha(\sqrt{I_1 * I_2}) - \frac{H_\alpha(I_1) + H_\alpha(I_2)}{2} \tag{2.25}$$

H_α représente l'entropie de Rényi avec α entier plus grand que 3 :

$$H_\alpha(I) = \frac{1}{1-\alpha} Log_2\Big(\sum_{k \in I} (g_I(k))^\alpha\Big) \tag{2.26}$$

Cependant, ces mesures n'ont pas produit de résultats particulièrement satisfaisants dans le cadre de l'évaluation de la segmentation [33, 35].

Métriques topologiques

Les mesures topologiques ont été proposées de façon à prendre en compte la position des pixels mal localisés afin de quantifier l'écart entre deux images. Elles semblent particulièrement intéressantes dans le cadre de l'évaluation de localisation puisque cela permet de considérer la distance entre les pixels mal localisés et la vérité terrain. Peli et Malah [39] utilisent deux mesures traduisant la distance entre les pixels mal localisés de l'objet et la vérité terrain :

$$DMoy(I_{vt}, I_l) = \frac{1}{Card(I_l^{Fr})} \sum_{k \in I_l^{Fr}} d(k, I_{vt}^{Fr}) \tag{2.27}$$

$$DMoC(I_{vt}, I_l) = \frac{1}{Card(I_l^{Fr})} \sum_{k \in I_l^{Fr}} d(k, I_{vt}^{Fr})^2 \tag{2.28}$$

avec $d(k_1, I) = min_{k_2 \in I} d(k_1, k_2)$.

Pratt [39, 40] a proposé une mesure empirique (Figure of Merit) entre les cartes de frontière I_l et de référence I_{vt}.

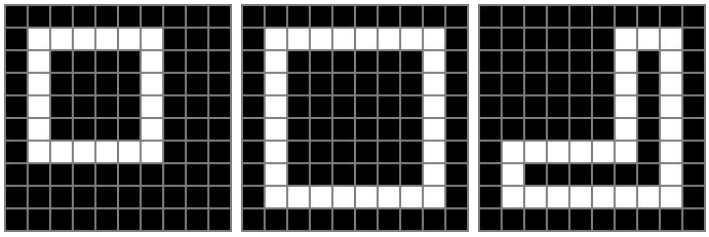
$$FOM(I_{vt}, I_l) = \frac{1}{MP} \sum_{k \in I_l^{Fr}} \frac{1}{1 + \alpha * d(k, I_{vt}^{Fr})^2} \tag{2.29}$$

avec MP correspondant à $Max(Card(I_{vt}^{Fr}), Card(I_l^{Fr}))$ et α une constante de normalisation. Un inconvénient de la mesure de Pratt est qu'elle n'est pas sensible aux erreurs de sous-détection alors qu'elle est sensible aux erreurs de sur-détection et de localisation. De plus, elle n'est pas sensible à la forme des zones erronées ce qui peut poser problème pour l'évaluation de la localisation d'un objet. Dans l'exemple présenté à la figure 2.20, la métrique de Pratt donne le même résultat d'évaluation pour les deux résultats de localisation. Cependant, l'étude comparative effectuée dans [33] montre qu'il s'agit de la métrique qui donne les meilleurs résultats pour l'évaluation de la segmentation.

La distance de Hausdorff représente une mesure de la distance spatiale entre deux ensembles de points, et dans le cas qui nous intéresse entre deux cartes de contours I_{vt} et I_l. Cette distance [41] est définie comme suit :

$$HAU(I_{vt}, I_l) = \max \left(\ h(I_{vt}, I_l) \ , \ h(I_l, I_{vt}) \ \right) \tag{2.30}$$

avec



(a) Vérité terrain (b) Premier résultat (c) Second résultat

FIGURE 2.20 – Limitation : la métrique de Pratt donne le même résultat d'évaluation pour les deux résultats de localisation.

$$h(I_1, I_2) = \max_{k_1 \in I_1^{Fr}} \min_{k_2 \in I_2^{Fr}} d(k_1, k_2) \tag{2.31}$$

où $d(k_1, k_2)$ représente la distance dans le plan, généralement la distance euclidienne, entre les points k_1 et k_2. Si $h(I_1, I_2) = \delta$, cela signifie que tous les points frontière de I_1 sont à une distance inférieure ou égale à δ d'au moins un point frontière de I_2. Si $HAU(I_{vt}, I_l) = d$, cela signifie que tous les points frontière de I_l sont à une distance inférieure ou égale à d d'au moins un point frontière de I_{vt} et vice versa. La distance de Hausdorff représente donc la distance entre le pixel de la localisation le plus éloigné et la vérité terrain.

Baddeley [34, 36] a proposé une variante de la distance de Hausdorff. Pour cela, il reformule tout d'abord :

$$HAU(I_{vt}, I_l) = \max_{k \in I_{vt}^{Fr} \cup I_l^{Fr}} \left(d(k, I_{vt}^{Fr}), d(k, I_l^{Fr}) \right) \tag{2.32}$$

avec $d(k, I^{Fr}) = \min_{k_2 \in I^{Fr}} d(k, k_2)$.

Il propose ensuite de remplacer l'opérateur max par une moyenne, ce qui aboutit à la formulation suivante de la distance de Baddeley :

$$BAD(I_{vt}, I_l) = \left[\frac{1}{card(I)} \sum_{k \in I_{vt}^{Fr} \cup I_l^{Fr}} |d(k, I_{vt}^{Fr}) - d(k, I_l^{Fr})|^P \right]^{\frac{1}{P}} \tag{2.33}$$

avec $P \geq 1$. On peut montrer que lorsque P tend vers l'infini, la distance de Baddeley tend vers la distance de Hausdorff.

Odet [42] a proposé une mesure de divergence échelonnable permettant d'évaluer différents niveaux d'erreur de résultats de segmentation binaires : évaluation de

détection de frontières avec un faible écart par rapport à la vérité terrain et erreur moyenne. Parmi les mesures de divergence proposées, on trouve les deux mesures suivantes :

$$ODI_n(I_{vt}, I_l) = \frac{1}{Card(I^{Fr}_{l\setminus vt})} \sum_{k \in I^{Fr}_{l\setminus vt}} \left(\frac{d(k, I^{Fr}_{vt})}{d_{Th}}\right)^n \tag{2.34}$$

$$UDI_n(I_{vt}, I_l) = \frac{1}{Card(I^{Fr}_{vt\setminus l})} \sum_{k \in I^{Fr}_{vt\setminus l}} \left(\frac{d(k, I^{Fr}_{l})}{d_{Th}}\right)^n \tag{2.35}$$

où d_{Th} correspond à une distance seuil et n correspond à un facteur d'échelle. La valeur de la distance de seuil d_{Th} dépend de l'application et fixe ce qui est considéré comme « loin ». On pourrait fixer cette distance seuil comme étant la plus grande distance entre deux vérités terrain réalisées par des personnes différentes afin de prendre en compte le manque de précision dans les vérités terrain. Tous les pixels sont pris en compte, même ceux qui sont loin de la référence, mais le paramètre d'échelle n permet de donner un poids différent aux pixels de la frontière proche de la référence et aux pixels de la frontière proche de la valeur du seuil. Choisir des faibles valeurs pour n $(0 < n < 1)$ donne plus d'importance aux pixels de la frontière très proches de la frontière de référence, ce qui conduit à une évaluation très précise. A l'opposé, choisir de grandes valeurs de n considère les pixels près de la frontière de référence comme corrects et amplifie juste les pixels de la frontière près de la distance seuil.

2.3.4 Évaluation de la localisation par des masques

La localisation d'un objet peut être représentée par un masque. L'évaluation de cette méthode de localisation peut s'apparenter à l'évaluation d'une méthode de segmentation en région. En effet, le fond correspondra alors à une première région et l'objet à une seconde. L'évaluation de la segmentation en région peut donc être transposée à l'évaluation de la localisation par masque.

Le projet Pascal VOC Challenge [43] utilise une métrique simple pour évaluer la localisation d'un seul objet :

$$PAS(I_{vt}, I_l) = \frac{Card(I^{Re}_{vt} \cap I^{Re}_{l})}{Card(I^{Re}_{vt} \cup I^{Re}_{l})} \tag{2.36}$$

où I^{Re}_{vt} correspond à l'ensemble des pixels appartenant à la région de l'objet d'intérêt dans l'image I_{vt}, $I^{Re}_{vt} \cap I^{Re}_{l}$ correspond à l'ensemble des pixels de l'objet correctement

localisés et $I_{vt}^{Re} \cup I_{l}^{Re}$ correspond à l'ensemble des pixels appartenant à l'objet ou reconnus comme tels. Cette métrique est comprise entre 0 et 1 et qualifie la qualité des pixels détectés de l'objet. Le résultat 1 correspond à un résultat optimal, c'est-à-dire lorsque $I_{vt}^{Re} \cap I_{l}^{Re} = I_{vt}^{Re} \cup I_{l}^{Re}$ soit $I_{vt}^{Re} = I_{l}^{Re}$.

Henricsson et Baltsavias [44] utilisent deux métriques pour évaluer leur travail de localisation. Ces deux métriques sont les suivants :

$$HEN1(I_{vt}, I_l) = \frac{|Card(I_{l}^{Re}) - Card(I_{vt}^{Re})|}{Card(I_{vt}^{Re})} \tag{2.37}$$

et

$$HEN2(I_{vt}, I_l) = \frac{Card(I_{l\setminus vt}^{Re}) + Card(I_{vt\setminus l}^{Re})}{Card(I_{vt}^{Re})} \tag{2.38}$$

Les métriques de Yasnoff *et coll.* [45] correspondent à des mesures d'erreur de classification lorsque les classes sont supposées connues. Les deux premières métriques sont basées sur la matrice de confusion CF. Cette matrice est construite en assignant à CF_{ij} le nombre de pixels de la classe j reconnus comme étant des pixels de classe i, avec i et $j \in [1, n]$, $n = \max(N^*(I_{vt}), N(I_l))$, $N^*(I_{vt})$ étant le nombre d'objets dans la vérité terrain et $N(I_l)$ le nombre d'objets dans le résultat de localisation. Les termes diagonaux CF_{ii} représentent alors les pixels correctement classés et les termes CF_{ij}, avec $i \neq j$, représentent les pixels mal classés. La première mesure d'erreur de Yasnoff pour la classe k est la suivante :

$$YAS_1(k) = 100 * \frac{\left(\sum_{i=1}^{n} CF_{ik}\right) - CF_{kk}}{\sum_{i=1}^{n} CF_{ik}} \tag{2.39}$$

$\sum_{i=1}^{n} CF_{ik}$ représente le nombre de pixels de la classe k, et CF_{kk} représente le nombre de pixels de la classe k correctement classés. La seconde mesure d'erreur de Yasnoff pour la classe k est la suivante :

$$YAS_2(k) = 100 * \frac{\left(\sum_{i=1}^{n} CF_{ki}\right) - CF_{kk}}{\left(\sum_{i=1}^{n}\sum_{j=1}^{n} CF_{ij}\right) - \sum_{i=1}^{n} CF_{ik}} \tag{2.40}$$

$\sum_{i=1}^{n} CF_{ki}$ représente le nombre de pixels étiquetés k, et $\left(\sum_{i=1}^{n}\sum_{j=1}^{n} CF_{ij}\right) - \sum_{i=1}^{n} CF_{ik}$ représente le nombre de pixels de l'image n'appartenant pas à la classe k. On peut ré-écrire les équations ainsi :

$$YAS_1(I_{vt}, I_l, k) = 100 * \frac{Card(I_l^{Re(k)}) - Card(I_{vt\cap l}^{Re(k)})}{Card(I_l^{Re(k)})} \tag{2.41}$$

$$YAS_2(I_{vt}, I_l, k) = 100 * \frac{Card(I_{vt}^{Re(k)}) - Card(I_{vt\cap l}^{Re(k)})}{Card(I) - Card(I_l^{Re(k)})} \tag{2.42}$$

Ces deux indicateurs permettent de rendre compte des erreurs de classification pour chaque classe, mais ne tiennent pas compte des informations spatiales sur les pixels mal classés. Afin de contrer ce problème, Yasnoff *et coll.* ont proposé un indicateur fondé sur le calcul d'un taux d'erreur pour un pixel mal classé a proportionnel à la distance entre ce pixel et le plus proche pixel appartenant à la classe à laquelle a aurait dû être affecté. L'erreur est alors définie comme suit :

$$YAS_3(I_{vt}, I_l, k) = \frac{100}{Card(I)} \sqrt{\sum_{a \in I_{l\setminus vt}^{Re(k)}} d(a, I_{vt}^{Re(k)})} \tag{2.43}$$

Wolf [46] a travaillé sur la définition de précision et de rappel au niveau du pixel. Ces deux métriques sont habituellement utilisées pour caractériser les performances globales d'un algorithme comme nous le verrons par la suite. Au niveau pixel, on a alors les définitions suivantes :

$$Ppx(I_{vt}, I_l) = \frac{Card(I_{vt}^{Re} \cap I_l^{Re})}{Card(I_l^{Re})} \tag{2.44}$$

$$Rpx(I_{vt}, I_l) = \frac{Card(I_{vt}^{Re} \cap I_l^{Re})}{Card(I_{vt}^{Re})} \tag{2.45}$$

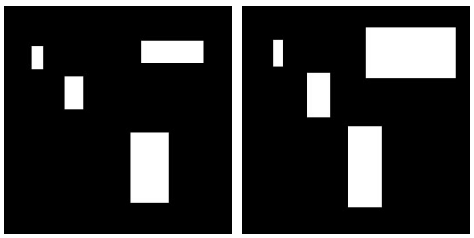
2.3.5 Évaluation de la localisation de plusieurs objets dans une image

Les métriques présentées jusqu'ici ne donnent pas de note globale lorsque plusieurs objets sont présents dans une image. Elles donnent un score pour chaque objet présent dans l'image. Dans le cas où la localisation est représentée par des masques et où plusieurs objets sont présents, nous avons alors le fond qui correspond à une région puis chaque objet correspond à une région supplémentaire. Ainsi, une image I_l, résultat de localisation, contient $N(I_l) + 1$ régions. Elle est égale à l'union des régions, $I_l^{Re(0)}$ correspondant au fond de l'image et $I_l^{Re(i)}$ correspondant à l'objet i.

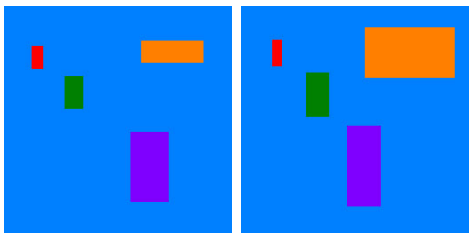
Certaines métriques, basées région, proposent de donner directement une note globale quel que soit le nombre d'objets présents dans la scène.

Mise en correspondance

La plupart des méthodes d'évaluation multi-objets effectuent une mise en correspondance des régions avant d'effectuer l'évaluation. Cette mise en correspondance consiste à différencier les objets présents dans une image, puis à identifier les objets qui se correspondent d'une image à l'autre. On peut voir dans la figure 2.21 deux images, une vérité terrain et un résultat de localisation, contenant plusieurs objets. Les régions de même couleur indiquent des objets mis en correspondance dans l'image de droite et dans celle de gauche.



(a) Vérité terrain et résultat de localisation



(b) Objets mis en correspondance

FIGURE 2.21 – Principe de la mise en correspondance

Cette mise en correspondance peut être effectuée de plusieurs façons mais on utilise souvent l'intersection maximale comme critère. Par ailleurs, on peut choisir de mettre une ou plusieurs fois une région en correspondance avec des régions de l'autre image si celles-ci maximisent l'intersection à chaque fois.

Généralisation d'une métrique à plusieurs objets

Une fois la mise en correspondance effectuée entre deux images, nous pouvons utiliser une métrique sur chacun des objets de l'image. Cependant, afin de donner

une note pour la localisation de l'ensemble des objets de l'image, nous devons adapter cette métrique pour la rendre multi-objets.

Supposons que nous ayons une métrique $M(I_{vt}, I_l, i)$ donnant une note de pertinence pour l'objet i dans l'image I_l par rapport à l'image référence I_{vt}. Si nous souhaitons la rendre multi-objets, il nous faut alors fusionner les résultats de chacun des objets. Cette fusion peut se faire de plusieurs façons. L'approche la plus simple pour rendre une métrique multi-objets est d'utiliser les opérateurs min ou max :

$$M(I_{vt}, I_l) = \min_{i \in [1, N(I)]} M(I_{vt}, I_l, i) \tag{2.46}$$

$$M(I_{vt}, I_l) = \max_{i \in [1, N(I)]} M(I_{vt}, I_l, i) \tag{2.47}$$

Une seconde approche consiste à faire une moyenne, éventuellement pondérée, des notes de chaque objet :

$$M(I_{vt}, I_l) = \frac{1}{i} \sum_{i \in [1, N(I)]} M(I_{vt}, I_l, i) * p(i) \tag{2.48}$$

où $p(i)$ est un coefficient de pondération. Ce coefficient peut, par exemple, tenir compte de la taille de l'objet dans l'image afin de donner plus d'importance aux objets les plus grands.

Les métriques multi-objets

Mis à part le fait de pouvoir adapter une métrique au cas multi-objets, certaines métriques ont été développées directement pour évaluer plusieurs objets. Ces métriques sont basées région.

Deux mesures d'erreur entre deux résultats de segmentation en région ont été définies par Martin [47]. Ces mesures permettent de donner une note globale quel que soit le nombre de régions. Pour cela, on utilise l'erreur de raffinement entre deux images I_1 et I_2 :

$$E(I_1, I_2, k) = \frac{Card(I_{1\backslash 2}^{Re(k)})}{Card(I_1^{Re(k)})}$$

où $Re(k)$ représente la région qui contient le pixel k. A partir de cette erreur de raffinement local, deux mesures globales, Global Consistency Error (GCE) et Local Consistency Error (LCE), ont été définies :

$$MAR_{gce}(I_{vt}, I_l) = \frac{1}{Card(I)} \min\big(\sum_{k\in I} E(I_{vt}, I_l, k), \sum_{k\in I} E(I_l, I_{vt}, k)\big) \tag{2.49}$$

$$MAR_{lce}(I_{vt}, I_l) = \frac{1}{Card(I)} \sum_{k\in I} \min(E(I_{vt}, I_l, k), E(I_l, I_{vt}, k)) \tag{2.50}$$

avec I le support commun de I_{vt} et I_l. Ces métriques ont été développées dans l'objectif de valider la création d'une base de résultats de segmentation expertes. Cependant, elles peuvent servir à comparer des résultats de segmentation entre eux, et plus particulièrement un résultat de segmentation quelconque avec une vérité terrain experte. On peut également envisager de les utiliser pour l'évaluation de la localisation de plusieurs objets dans une image. D'après les études menées sur ces deux métriques [47], MAR_{lce} est plus pénalisant que MAR_{gce} et donne de meilleurs résultats.

La distance de Hamming [48] est également l'une des méthodes d'évaluation de segmentation en région. Elle se calcule après une mise en correspondance des régions afin que les régions de même indice aient un recouvrement maximal. On peut alors calculer la distance de Hamming directionnelle suivante :

$$D_H(I_{vt}, I_l) = \sum_{i=0}^{N(I_l)} \sum_{j=0, j\neq i}^{N^*(I_{vt})} Card\big(I_l^{Re(i)} \cap I_{vt}^{Re(j)}\big) \tag{2.51}$$

Le terme central est la somme, pour la région $I_l^{Re(i)}$, des aires de toutes les intersections non maximales et non vides avec les régions de I_{vt}. A partir de cette expression, la distance normalisée de Hamming est définie comme suit :

$$HAM(I_{vt}, I_l) = 1 - \frac{D_H(I_{vt}, I_l) + D_H(I_l, I_{vt})}{2 * Card(I)} \tag{2.52}$$

La distance de Hamming est comprise entre 0 et 1, un résultat proche de 1 correspond à une forte mise en correspondance entre le résultat de l'algorithme et la vérité terrain.

Dans [49], Hafiane définit une métrique d'évaluation d'un résultat de segmentation en région. Cette métrique effectue également une mise en correspondance, avec la possibilité de mettre en correspondance plusieurs fois une même région :

$$HAF1(I_{vt}, I_l) = \eta \sum_{i, \mathrm{argmax}_j\, Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})} \frac{Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})}{Card(I_{vt}^{Re(i)} \cup I_l^{Re(j)})} \tag{2.53}$$

avec

$$\eta = \begin{cases} \frac{N^*(I_{vt})}{N(I_l)} & siN(I_l) \geq N^*(I_{vt}) \\ \\ \frac{1}{Log\left(\frac{N^*(I_{vt})}{N(I_l)}\right)} & autrement \end{cases}$$

Cette métrique semble intéressante dans la mesure où le terme central de la somme correspond au critère d'évaluation utilisé par le Pascal VOC Challenge. Ainsi, cet métrique est une généralisation à plusieurs objets du critère PAS. Hafiane *et coll.* [50] ont ensuite proposé une amélioration :

$$M(I_{vt}, I_l) = \sum_{j,\text{argmax}_i\, Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})} \frac{Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})}{Card(I_{vt}^{Re(i)} \cup I_l^{Re(j)})}\, \rho_j \tag{2.54}$$

avec :

$$\rho_j = \frac{Card(I_l^{Re(j)})}{Card(I)} \tag{2.55}$$

La valeur ρ_j permet de pondérer l'influence de la région j en fonction de sa taille dans l'image. Par ailleurs, le paramètre η, qui permet de pénaliser en cas de sur-segmentation ou de sous-segmentation, a été modifié par rapport à la première métrique :

$$\eta = \begin{cases} \frac{N^*(I_{vt})}{N(I_l)} & siN(I_l) \geq N^*(I_{vt}) \\ \\ log(1 + N(I_l)/N^*(I_{vt})) & sinon \end{cases} \tag{2.56}$$

La métrique finale est alors le suivant :

$$HAF2(I_{vt}, I_l) = \frac{M(I_{vt}, I_l) + m \times \eta}{1 + m} \tag{2.57}$$

la valeur m permet de pondérer la pénalisation de la sur- et de la sous-segmentation.

Le distance de Vinet [51, 52] permet de calculer une mesure de dissimilarité entre deux classifications. Aucune hypothèse sur les classes n'est nécessaire puisque l'algorithme effectue une mise en correspondance. Cette distance peut être utilisée pour comparer un résultat de localisation de plusieurs objets à une vérité terrain. On construit la table de superposition suivante :

$$T(I_{vt}, I_l) = \left[Card(I_l^{Re(i)} \cap I_{vt}^{Re(j)}) \ ,\ i = 1..N(I_l), j = 1..N^*(I_{vt}) \right] \tag{2.58}$$

où $Card(I_l^{Re(i)} \cap I_{vt}^{Re(j)})$ est le nombre de pixels étiqueté i dans I_l en correspondance avec les pixels étiquetés j dans I_{vt}.

Avec cette table, on recherche les classes appariées de manière récursive :

1. dans le tableau, on sélectionne les deux classes i et j qui maximisent l'intersection,
2. tous les éléments du tableau qui font partie de la ligne et de la colonne de la cellule sélectionnée sont mis à 0,
3. tant qu'il reste des éléments, on retourne à la première étape.

Avec les cellules sélectionnées, Vinet donne une mesure de dissimilarité. Soit C' l'ensemble des cellules sélectionnées :

$$VIN(I_{vt}, I_l) = Card(I) - \sum_{C'} Card(I_l^{Re(i)} \cap I_{vt}^{Re(j)}) \tag{2.59}$$

La table de superposition permet de faire une mise en correspondance par maximisation de l'intersection des régions. Cependant, contrairement à la métrique de Hafiane, une région ne peut être mise en correspondance qu'une seule fois. Si l'on suppose que les classes sont connues et déjà mises en correspondance, alors il n'est pas nécessaire d'effectuer la mise en correspondance par la métrique critère de Vinet. La table de superposition est alors égale à la matrice de confusion CF, comme dans le cas de la métrique de Yasnoff. La métrique de Vinet est alors égale à :

$$VIN(I_{vt}, I_l) = Card(I) - \sum_{i}^{N(I)} CF_{ii} \tag{2.60}$$

$$VIN(I_{vt}, I_l) = Card(I) - \sum_{i}^{N(I)} Card(I_{vt \cap l}^{Re(i)}) \tag{2.61}$$

Cette mesure donne de bons résultats dans le cadre de l'évaluation de la segmentation [33].

2.4 Métriques d'évaluation d'un résultat de reconnaissance

Nous avons vu qu'il était possible de donner une mesure de pertinence à un résultat de localisation. Nous souhaitons faire de même avec un résultat de reconnaissance. Cependant, il n'est pas possible de calculer directement une distance sur les labels renvoyés par les algorithmes, c'est-à-dire sur les identifiants numériques

ou les chaînes de caractères renvoyés. En effet, ces identifiants sont des variables qualitatives non ordonnées, et calculer une distance entre ces variables n'aurait pas de sens. Naïvement, la seule méthode permettant de calculer une distance sur ces variables est de regarder si celles-ci sont égales, et donc de dire que la distance est de 0 si les identifiants sont identiques, et 1 sinon.

La méthode basique n'étant pas satisfaisante, nous allons nous intéresser aux méthodes permettant de calculer une distance entre deux représentations des objets, et plus particulièrement aux ensembles de points et aux graphes. L'évaluation de la reconnaissance consistera alors à valuer la distance entre la description de l'objet renvoyée par l'algorithme et la description de la classe affectée dans la base de données. A titre d'exemple, nous souhaitons que la distance entre les classes « chat » et « chien » soit plus petite que la distance entre les classes « voiture » et « chien ».

2.4.1 Distance entre ensembles de points

Avant de calculer une distance entre deux ensembles de points, il faut pouvoir calculer une distance entre deux points, c'est-à-dire entre vecteurs. On peut alors s'intéresser aux distances usuelles entre deux vecteurs dans $\mathbb{R}^n$. Si l'on note $X = (x_1, x_2, \ldots, x_n)$ le premier vecteur et $Y = (y_1, y_2, \ldots, y_n)$ le second, alors on a les distances suivantes :

– Distance de Manhattan :

$$D_1(X,Y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.62}$$

– Distance euclidienne :

$$D_2(X,Y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{2.63}$$

– Distance de Minkowski :

$$D_p(X,Y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \tag{2.64}$$

– Distance de Tchebychev :

$$D_\infty(X,Y) = \lim_{p\to\infty} \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} = \sup_i |x_i - y_i| \tag{2.65}$$

De plus, on peut également utiliser la distance de Hausdorff ou bien la distance de Hamming que l'on a déjà présentées dans la section précédente. Toutes ces distances peuvent éventuellement être utilisées dans le cas d'une représentation des objets par des sacs de mots.

2.4.2 Distance entre graphes

La distance entre deux graphes peut être vue comme la différence qui réside entre les deux graphes G_1 et G_2. Cependant, afin de pouvoir calculer cette distance, il faut auparavant apparier ces deux graphes. Dans le cas de l'interprétation d'images, on s'intéresse à un appariement inexact de graphes. Ce problème consiste à assigner les nœuds du graphe G_1 à ceux du graphe G_2 tout en minimisant le coût. Ce problème peut être notamment résolu par la distance d'édition, l'algorithme hongrois, la distance de Bunke...

Distance d'édition

L'appariement de deux graphes peut être vu comme l'édition du premier graphe G_1 pour le transformer en un graphe isomorphe au graphe G_2. Cette édition se fait en une succession d'éditions élémentaires :

- la substitution d'un nœud $x \in V_1$ par un nœud $y \in V_2$, noté $x \to y$,
- l'insertion d'un nœud, noté $\epsilon \to y$,
- la suppression d'un nœud, noté $x \to \epsilon$.

Une succession d'éditions élémentaires $s_1, \ldots, s_p$ permet de passer de G_1 à G_2. Cette succession s'appelle une séquence d'édition S. Parmi l'ensemble des séquences possibles, il y en a une qui a un coût minimum. Cette séquence permet de définir la distance d'édition :

$$D_{\text{édition}}(G_1, G_2) = \sum_{i=1}^{p} c(s_i) \tag{2.66}$$

où $c(s_i)$ représente le coût de l'édition s_i. Le coût associé à la substitution, l'insertion ou la suppression d'un nœud dépend de l'application visée. Cependant, il est courant que l'on impose que la règle d'insertion et de suppression ait un coût supérieur à la substitution. Le principal inconvénient de cette méthode réside dans sa complexité en temps qui est exponentielle. De fait, cette méthode n'est alors pas envisageable sur des graphes trop grands.

Algorithme de Munkres

L'algorithme de Munkres [53], également appelé algorithme hongrois, permet de calculer une approximation de la distance d'édition. Cependant, cette méthode ne prend pas en compte le coût d'édition pour les arêtes, mais uniquement celui pour la substitution des nœuds. Avant d'utiliser cet algorithme, il faut définir une matrice des coûts $MC = (mc_{i,j})$. Les éléments $mc_{i,j}$ de cette matrice correspondent aux coûts nécessaires pour substituer le nœud $v_i \in V_1$ par le nœud $u_j \in V_2$, c'est-à-dire $mc_{i,j} = c(v_i \to u_j)$. Dans le cas de l'interprétation d'images, ce coût peut être, par exemple, la distance entre le vecteur caractéristique du nœud v_i et celui du nœud u_j.

Une fois cette matrice calculée, l'algorithme de Munkres permet de trouver l'assignation $A = a_1, \ldots, a_n$ optimale minimisant le coût global. Ce coût global est une approximation de la distance d'édition :

$$D_{\text{Munkres}}(G_1, G_2) = \sum_{i=1} mc(i, a_i) \tag{2.67}$$

Afin de trouver l'assignation A, l'algorithme de Munkres [53, 54] utilise la matrice des coûts. Tout d'abord, pour chaque ligne de cette matrice, on soustrait la valeur minimale de cette ligne. On obtient alors au moins une cellule avec un zéro par ligne, qui correspond à l'assignement optimal des nœuds correspondants. Il faut alors vérifier que chaque nœud de G_1 est assigné à un seul nœud de G_2. On regarde pour cela si chaque colonne a au moins un zéro. Si c'est le cas, alors l'assignement optimal se trouve parmi les cellules ayant un zéro. Si une colonne ne possède pas de zéro, cela signifie que le nœud correspondant ne peut être assigné de manière optimale. Dans ce cas, on marque toutes les colonnes et les lignes ayant un zéro, puis on cherche un assignement optimal parmi les colonnes et les lignes non marquées de manière récursive.

Algorithme de Bunke

L'algorithme de Bunke *et coll.* [54] est une amélioration de l'algorithme hongrois permettant de prendre en compte le coût d'édition des arêtes. Cette méthode permet d'avoir une meilleure estimation de la distance d'édition tout en gardant une complexité raisonnable. Le principe de l'algorithme de Bunke est de calculer une matrice de coûts similaire à celle utilisée par l'algorithme hongrois. Cependant, au lieu de calculer uniquement le coût pour substituer le nœud $v_i \in V_1$ par le nœud $u_j \in V_2$, on calculera en plus le coût minimal nécessaire pour substituer toutes les arêtes connectées à v_i par les arêtes connectées à u_j. Ainsi, le coût sera le suivant :

$$mc_{i,j} = c(v_i \rightarrow u_j) + \min\{\sum c(e_{vi} \rightarrow e_{uj})\} \tag{2.68}$$

où e_{vi} correspond à l'ensemble des arêtes connectées à v_i. Pour calculer $\min\{\sum c(e_{vi}) \rightarrow e_{uj}\}$, on utilisera également l'algorithme de Munkres avec une matrice des coûts dont les éléments correspondent aux coûts pour substituer les arêtes connectées à v_i par les arêtes connectées à u_j.

D'autres algorithmes permettent de calculer une distance entre deux graphes. On peut notamment citer les algorithmes de Bunke et Shearer [55] ou de Fernandez et Valiente [56]. Ces distances sont basées sur le calcul du sous-graphe commun maximal, mais semblent peu applicables dans le cas de la reconnaissance d'objet.

2.5 Métriques d'évaluation d'un résultat d'interprétation d'images

Nous avons vu précédemment comment nous pouvons obtenir une note pour un résultat de localisation ou de reconnaissance d'un ou plusieurs objets dans une image. A partir de cela, nous pouvons donner une note ou une décision finale afin de dire si une image est bien interprétée, c'est-à-dire si les objets présents dans l'image ont bien été détectés, localisés et reconnus.

Disposant de plusieurs métriques permettant de donner une note à un résultat d'interprétation d'image, c'est-à-dire à la localisation et/ou la reconnaissance d'un ou plusieurs objets dans une image, l'objectif est ici de donner une note finale ou une décision finale concernant le résultat de cette image, afin de dire si l'algorithme a renvoyé un résultat correct. Une note finale peut être comprise, par exemple, entre 0 et 1, tandis qu'une décision finale prendra la forme vrai/faux.

2.5.1 Normalisation des données

Avant de pouvoir obtenir un score global, il peut être nécessaire de normaliser les données. En effet, les scores provenant de différentes métriques d'évaluations ne sont pas forcément dans le même intervalle. Une métrique peut être dans [0 ∞[et une autre dans [0 1]. Il faut donc les normaliser afin que les données soient dans le même intervalle et que l'on puisse les combiner sans qu'une donnée prennent plus d'importante que les autres.

La première méthode proposé dans [57] est la normalisation $\min - \max$:

$$s'_k = \frac{s_k - \min}{\max - \min} \tag{2.69}$$

où s_k représente la série de données à normaliser (par exemple, l'ensemble de résultats d'évaluation de localisation). L'inconvénient de cette méthode est que la normalisation des données dépend des résultats d'évaluation.

Une autre méthode est la mise à l'échelle décimale [57]. Cette méthode permet de normaliser des données qui seraient sur des échelles logarithmiques différentes. Par exemple, si les résultats d'évaluation de localisation $s1_k$ sont compris dans $[0, 1000]$ et les résultats d'évaluation de reconnaissance $s2_k$ sont compris dans [0 1], on pourra appliquer la normalisation suivante :

$$s'_k = \frac{s_k}{10^n} \tag{2.70}$$

avec $n = \log_{10} \max(s_k)$. Cette méthode présente le même inconvénient que la normalisation $\min - \max$.

La méthode la plus utilisée est le z-score. Pour cette méthode, on a besoin d'avoir la moyenne ainsi que l'écart type des données. L'avantage de cette méthode est que l'on peut avoir une connaissance *a priori* de ces données et ainsi la normalisation ne dépend pas de la série s_k. Cette normalisation est la suivante :

$$s'_k = \frac{s_k - \mu}{\sigma} \tag{2.71}$$

Cette méthode est particulièrement efficace dans le cas où les données des séries s_k suivent une loi gaussienne.

Enfin, des méthodes plus élaborées permettent de normaliser les données avec plus d'efficacité. Cappelli *et coll.* [58] proposent d'utiliser une fonction double sigmoïde pour normaliser des données :

$$s'_k = \begin{cases} \frac{1}{1+\exp(-2(\frac{s_k - t}{r_1}))} & \text{si } s_k < t \\ \frac{1}{1+\exp(-2(\frac{s_k - t}{r_2}))} & \text{sinon} \end{cases} \tag{2.72}$$

avec t une valeur de référence et r_1, r_2 deux bornes telles que la fonction soit linéaire dans l'intervalle $[t - r_1, t - r_2]$. Cette normalisation présente de nombreux avantages, principalement par sa capacité à changer de comportement en fonction des valeurs assignées à r_1 et r_2.

2.5.2 Moyenne

Une méthode qui peut être employée afin d'obtenir un score global est d'utiliser une moyenne. Si la moyenne arithmétique peut être utilisée, la méthode utilisée le plus souvent est le F-score, qui est une moyenne harmonique entre deux éléments. Cette méthode est, par exemple, utilisée par Wolf [46] et le projet ETISEO [59] afin d'obtenir une note moyenne entre la précision (Pr) et le rappel (Ra) (la précision et le rappel sont détaillés page 62) :

$$Fscore = 2\frac{\mathrm{Pr} * \mathrm{Ra}}{\mathrm{Pr} + \mathrm{Ra}} \tag{2.73}$$

On peut également généraliser cette mesure au calcul du score F_β, dont le Fscore est un cas particulier avec $\beta = 1$:

$$F_\beta = (1 + \beta^2)\frac{\mathrm{Pr} * \mathrm{Ra}}{\beta^2 * \mathrm{Pr} + \mathrm{Ra}} \tag{2.74}$$

2.5.3 Seuillage

Le seuillage permet d'obtenir une décision finale concernant l'interprétation d'une image. Cela consiste à vérifier si un ou plusieurs résultats de localisation et/ou de reconnaissance sont inférieurs ou supérieurs à une valeur seuil λ.

Dans le projet Robin [2], une fonction de décision G est définie afin de donner une décision finale sur la localisation d'un objet. Cette fonction de décision utilise les trois métriques définies précédemment :

$$G(I_{vt}, I_l) = \begin{cases} 1 \text{ si } ROB_{loc}(I_{vt}, I_l) \leqslant \varepsilon_1 \text{ et } ROB_{com}(I_{vt}, I_l) \leqslant \varepsilon_2 \text{ et } ROB_{cor}(I_{vt}, I_l) \leqslant \varepsilon_3 \\ 0 \text{ autrement} \end{cases} \tag{2.75}$$

La validation d'une bonne localisation repose sur la définition de ces seuils $\varepsilon_1, \varepsilon_2$ et ε_3. Deux valeurs ont été définies pour ces seuils afin d'évaluer leur influence :

- tolérance faible : $\varepsilon_1 = 0.15$, $\varepsilon_2 = 0.5$, $\varepsilon_3 = 0.15$
- tolérance stricte : $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.2$, $\varepsilon_3 = 0.05$

La fonction de décision G consiste donc bien à faire un seuillage sur les trois notes obtenues, puis à renvoyer un résultat 1 si chacune des notes est inférieure à un seuil.

2.5.4 Fusion

Il est également possible de fusionner plusieurs résultats de métriques différents pour obtenir une note ou une décision finale [33]. Cette fusion peut se faire de

différentes façons : fusion par combinaison linéaire, floue, par apprentissage ... La fusion par combinaison linéaire revient à faire une moyenne pondérée des différentes métriques. La fusion par apprentissage va permettre d'obtenir une décision finale concernant la qualité du résultat d'interprétation. Pour cela, des méthodes d'apprentissage vont être utilisées tels que les réseaux de neurones [60] et les séparateurs à vaste marge [61].

2.6 Métriques d'évaluation globale d'un algorithme d'interprétation d'images

Nous avons vu jusqu'à présent comment obtenir une note d'évaluation pour un résultat d'interprétation. Cependant, l'évaluation sur une image ne représente pas les performances globales de l'algorithme. Afin d'évaluer ces performances, il est nécessaire de faire une évaluation sur un grand ensemble d'images. L'utilisation de bases de données permet cette évaluation à grande échelle avec des conditions d'acquisitions différentes. Il est possible de réaliser une évaluation supervisée ou une évaluation non supervisée.

2.6.1 Évaluation supervisée

L'évaluation supervisée de l'interprétation requiert souvent la définition de vrais positifs, faux positifs et faux négatifs. Ces définitions sont les suivantes :

- Vrai positif (VP) : un vrai positif correspond à la bonne interprétation d'une image, c'est-à-dire lorsque l'algorithme renvoie une localisation et/ou une classe en accord avec la vérité terrain,
- Faux positif (FP) : un faux positif correspond à une mauvaise interprétation de l'image, c'est-à-dire lorsque l'algorithme interprète l'objet en désaccord avec la vérité terrain ou interprète un objet alors qu'il n'y en a pas dans la vérité terrain,
- Faux négatif (FN) : un faux négatif correspond à une absence d'interprétation de l'image, c'est-à-dire que l'algorithme ne renvoie pas de localisation et/ou de classe alors qu'il y a un objet.

Étant données ces définitions, nous disposons de plusieurs méthodes d'évaluation globale. La méthode la plus courante consiste à construire une courbe ROC. Une autre méthode également très utilisée est la construction de la courbe de Précision/Rappel.

Enfin, l'utilisation des matrices de confusion ou de tests du χ^2 est plus rare bien qu'elle permette d'obtenir de nombreux renseignements également.

Courbe ROC

L'une des méthodes les plus souvent employées afin d'évaluer globalement un algorithme d'interprétation est la méthode ROC (Receiver Operating Characteristic) [62, 43, 63, 64]. Le but est de tracer une courbe représentant le nombre de vrais positifs en fonction du nombre de faux positifs pour les différents paramétrages de l'algorithme. Afin de normaliser les courbes ROC, on représente généralement le taux de vrais positifs en fonction du taux de faux positifs. Ces taux sont définis ainsi [62] :

- Taux de vrais positifs :

$$T_{VP} = \frac{Card(VP)}{Card(VP) + Card(FN)} \tag{2.76}$$

- Taux de faux positifs :

$$T_{FP} = \frac{Card(FP)}{Card(VP) + Card(FP)} \tag{2.77}$$

Afin de pouvoir tracer la courbe, on fait varier un paramètre de l'algorithme. Pour chaque valeur de ce paramètre, on relève alors le nombre de vrais positifs et de faux positifs qu'on place sur la courbe. En prenant suffisamment de valeurs pour ce paramètre, on obtient une courbe telle que celles présentées dans la figure 2.22. Afin de comparer les différents algorithmes, on peut également utiliser deux mesures déduites des courbes ROC :

- le taux d'égales erreurs (EER : Equal Error Rate) :
  L'EER correspond au taux d'erreur obtenu lorsque le taux de faux négatifs est égal au taux de faux positifs. Ce taux de faux négatifs n'est pas lisible directement sur la courbe, mais est cependant accessible car égal à $1 - T_{VP}$. Plus l'EER est petit, plus l'algorithme est performant.
- l'aire sous la courbe (AUC : Area Under Curve) :
  L'AUC correspond, comme son nom l'indique, à l'aire sous la courbe ROC. L'AUC a l'avantage de mieux représenter les performances globales de l'algorithme. Plus l'AUC est grande, plus l'algorithme est performant.

On peut voir sur la figure 2.23 comment retrouver ces mesures à partir de la courbe ROC. Ainsi, l'EER correspond à l'abcisse de l'intersection entre la courbe et la diagonale décroissante. On peut également voir l'AUC sur la figure 2.23. En pratique,

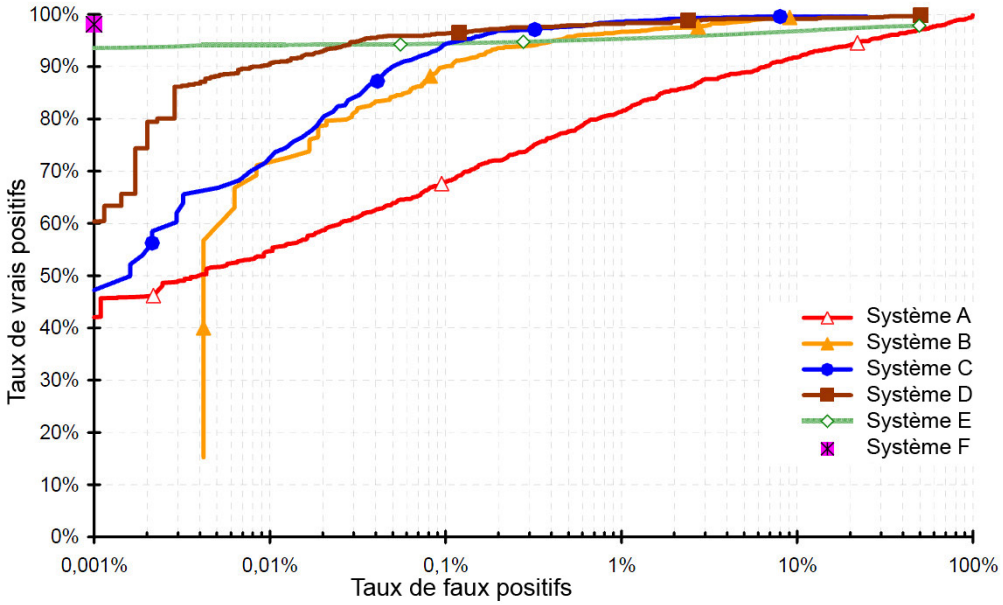


FIGURE 2.22 – Exemple de courbes ROC [64]

un algorithme qui sera bien classé avec l'EER sera, en général, bien classé avec l'AUC.

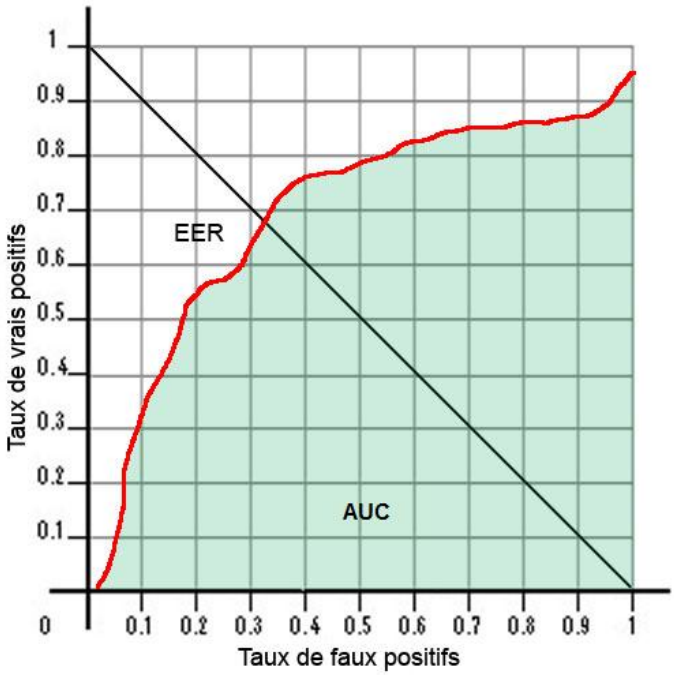


FIGURE 2.23 – Exemple de courbes ROC avec EER et AUC

Une autre représentation des courbes ROC consiste à représenter $1 - T_{VP}$ en fonction de T_{FP}. Cette courbe est alors appelée courbe DET et permet de mieux

visualiser les deux types d'erreurs possibles. De fait, de telles courbes sont beaucoup utilisées dans le domaine de la biométrie par exemple. Les courbes DET de la figure 2.24 représentent les mêmes données que les courbes ROC de la figure 2.22.

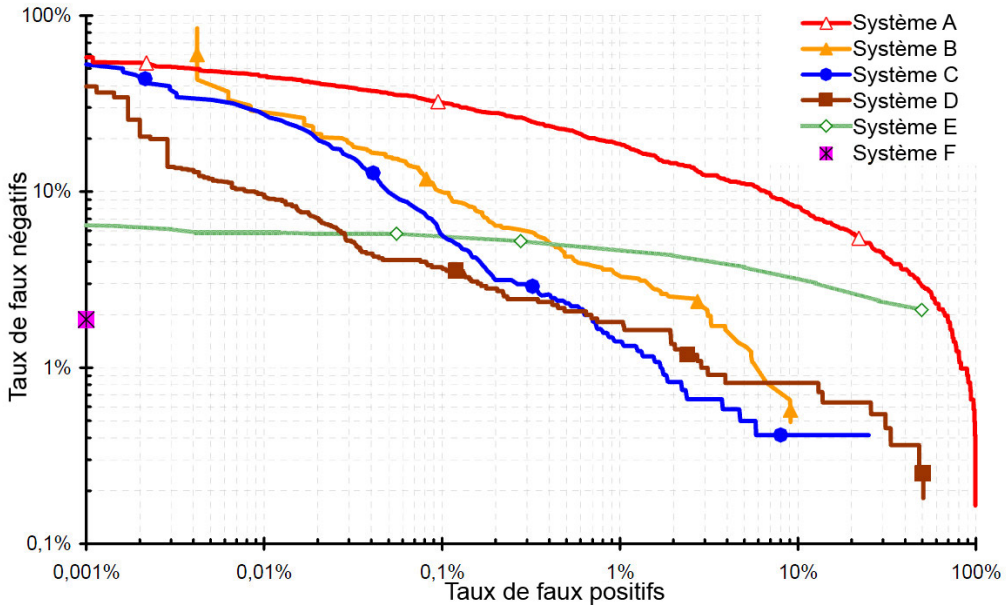


FIGURE 2.24 – Exemple de courbes DET [64]

Courbe Précision/Rappel

Une autre méthode d'évaluation régulièrement rencontrée est la méthode Précision/Rappel [43, 65]. Cette méthode est généralement utilisée afin d'évaluer les algorithmes de détection et de localisation. Le but est cette fois-ci de représenter la précision en fonction du rappel. La précision et le rappel sont définis comme suit :

- Précision (Pr) : la précision correspond au nombre de résultats corrects sur le nombre de positifs renvoyé par l'algorithme. On a donc ceci :

$$Pr = \frac{Card(VP)}{Card(VP) + Card(FP)} \tag{2.78}$$

- Rappel (Ra) : le rappel correspond au rapport entre le nombre de résultats corrects renvoyé par l'algorithme et le nombre de résultats corrects attendu (c'est-à-dire présent dans la vérité terrain) :

$$Ra = \frac{Card(VP)}{Card(VP) + Card(FN)} \tag{2.79}$$

On peut voir que le rappel est égal au taux de vrais positifs (T_{VP}).

Afin de pouvoir tracer la courbe, on fait varier un paramètre de l'algorithme tout comme avec la méthode ROC. Pour chaque valeur de ce paramètre, on relève alors la précision et le rappel que l'on place sur la courbe. En prenant suffisamment de valeurs pour ce paramètre, on obtient une courbe telle que celle présentée à la figure 2.25.

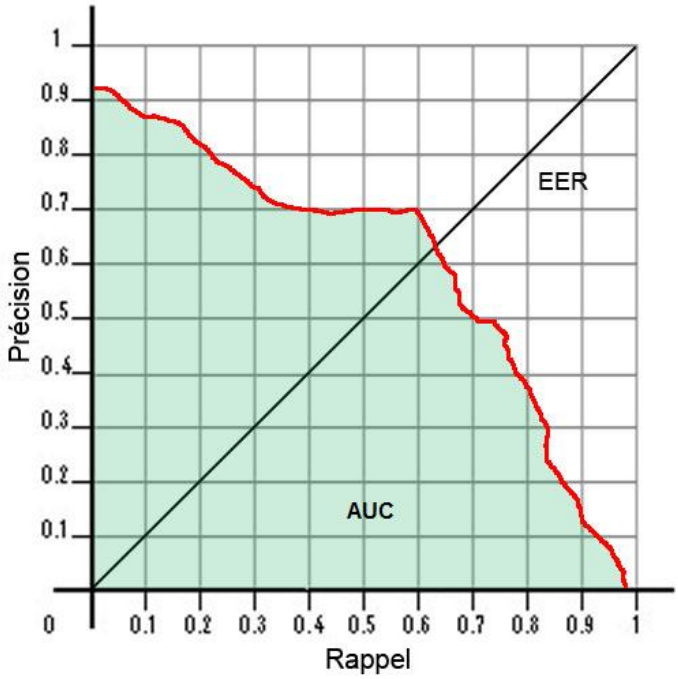


FIGURE 2.25 – Exemple de courbe Pr/Ra avec l'EER et l'AUC

On peut lire sur la courbe de précision/rappel l'EER et l'AUC comme avec la méthode ROC. L'EER se lit sur la figure 2.25 à l'intersection entre la courbe Pr/Ra et la diagonale croissante. L'EER trouvé ainsi est égal à l'EER trouvé avec la méthode ROC. L'AUC, qui n'est pas égal à celui de la courbe ROC, permet également de bien représenter les performances globales de l'algorithme. Plus l'AUC est grand, meilleures sont les performances de l'algorithme.

On peut également définir différents critères [43, 65] :
- le F-score (voir page 58),
- la précision pour le rappel max : P^*,
- le rappel pour la précision max : R^*,
- le rappel pour une précision de 0.5,
- la précision moyenne interpolée : elle correspond à la moyenne de 11 valeurs de la courbe Pr/Ra et se calculent ainsi (voir Fig. 2.26) :

$$Pm = \frac{1}{11} \sum_{Ra \in \{0, 0.1, \ldots, 1\}} Pr_{interp}(Ra) \tag{2.80}$$

avec

$$Pr_{interp}(Ra) = \max_{\widetilde{Ra} \geq Ra} Pr(\widetilde{Ra}) \tag{2.81}$$

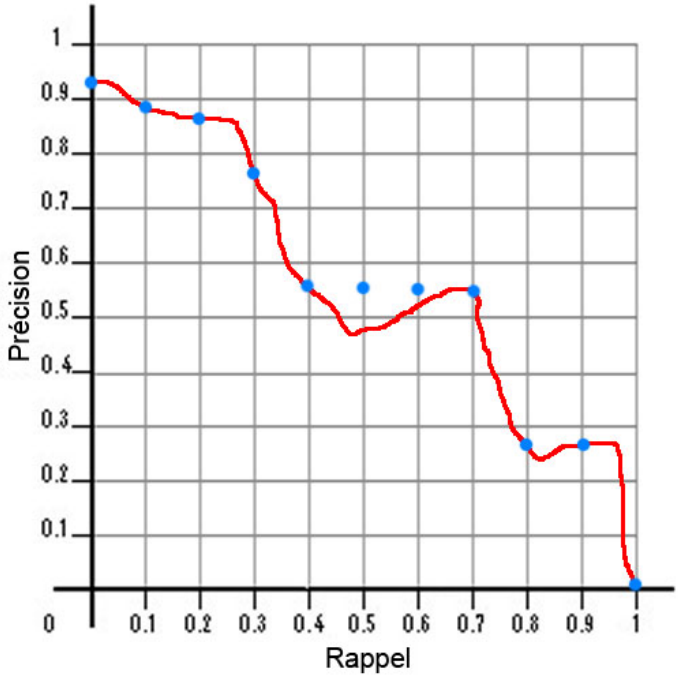


FIGURE 2.26 – Calcul de la précision moyenne interpolée

Matrice de confusion

Le projet ROBIN [2] évalue le problème de détection et de reconnaissance à partir de matrices de confusion. Le but est de faire une matrice carrée η avec autant de lignes que de classes présentes dans la base de données. A chaque fois que l'algorithme reconnaît l'objet i et que l'objet j est présent dans la vérité terrain, on incrémente la case (i, j). Une ligne est également prévue pour les classes *Autre* et *Ambigu*. Les vrais positifs sont alors sur la diagonale de la matrice. Les cases de la matrice sont définies ainsi :

$$\eta(i, j, \lambda) = \frac{1}{N^*(j)} \sum_{n \in B} \mathbf{1}_i(Y^n_{alg}) \cdot \mathbf{1}_j(Y^n_{vt}) \tag{2.82}$$

où B est la base de données de test, n un objet dans la base, i, j les classes de la base, $\mathbf{1}$ la fonction indicatrice, Y^n_{vt} correspond à la classe de l'objet n dans la vérité terrain,

Y_{alg}^n la classe de l'objet n renvoyée par l'algorithme (en fonction du paramètre λ) et $N^*(c)$ le nombre d'objets de la classe c dans la base, c'est-à-dire :

$$N^*(c) = \sum_{n \in B} \mathbf{1}_c(Y_{vt}^n) \tag{2.83}$$

La fonction indicatrice $\mathbf{1}_c(Y_{vt}^n)$ renvoie 1 si la classe de l'objet n est égale à c dans la vérité terrain, et 0 sinon.

Le projet ROBIN définit un premier critère de discrimination $D(\lambda)$, qui va correspondre à la somme pondérée des éléments présents sur la diagonale de la matrice de confusion :

$$D(\lambda) = \sum_{c \in \mathcal{Y}} \pi(c) \cdot \eta(c, c, \lambda) \tag{2.84}$$

avec $\pi(c)$ le coefficient de pondération correspondant au rapport du nombre d'élément de la classe c sur le nombre d'éléments de la base de données :

$$\pi(c) = \frac{N^*(c)}{\sum_{i \in \mathcal{Y}} N^*(i)} \tag{2.85}$$

De même, un critère d'indécision est défini. Cette fois-ci, on va s'intéresser à la classe $Am = Ambigu$. Le critère $I(\lambda)$ est défini ainsi :

$$I(\lambda) = \sum_{c \in \mathcal{Y}} \pi(c) \cdot \eta(Am, c, \lambda) \tag{2.86}$$

On peut alors définir plusieurs taux pour comparer les différents algorithmes :
- la discrimination pour l'incertitude minimum : D^*,
- l'incertitude pour la discrimination maximale : I^*,
- le taux d'incertitude et de discrimination égales : EDI.

On définit également un critère de rejet, qui dénote la capacité de l'algorithme à détecter une classe $Au = Autre$. Ce critère est défini comme suit :

$$R(\lambda) = \frac{1}{N^*(Au)} \sum_{n \in B} \mathbf{1}_{Au}(Y_{alg}^n) \cdot \mathbf{1}_{Au}(Y_{vt}^n) \tag{2.87}$$

Test d'adéquation du χ^2

Le test du χ^2 [66] est un test statistique permettant d'évaluer une corrélation entre deux ensembles de données numériques. Ce test peut être utilisé de différentes façons : test d'adéquation, test d'homogénéité et test d'indépendance. Le test d'adéquation

peut permettre de faire de l'évaluation supervisée d'algorithme de reconnaissance.

Ce test consiste à juger de l'adéquation entre la série de données statistiques et une loi de probabilité supposée suivie par la série statistique. Dans le cas de l'évaluation de reconnaissance, la loi de probabilité est en réalité donnée par la vérité terrain. Elle est construite ainsi :

$$S_{vt} = \{N^*(c)|c \in \mathcal{Y}\}$$

où $N^*(c)$ correspond au nombre de fois où la classe c est présente dans l'ensemble de la base de données. De même, on peut créer la série de données statistiques S_{alg} correspondant aux données $N(c)$ renvoyées par l'algorithme.

Le test du χ^2 de Pearson consiste alors à calculer l'indice suivant :

$$CHI2_{Pearson}(S_{vt}, S_{alg}) = \sum_{c \in \mathcal{Y}} \frac{(N(c) - N^*(c))^2}{N^*(c)} \tag{2.88}$$

On estime que le résultat de l'algorithme est bon si le résultat de $CHI2_{Pearson}(\cdot, \cdot)$ est inférieur à une valeur seuil χ_{Th}. Cette valeur seuil est calculée à partir du degré de liberté du système $k = N^*(\mathcal{Y}) - 1$ et de l'erreur Err généralement fixée à 5%. On utilise alors la loi du χ^2 dont la densité de probabilité est définie par $f_\chi(t)$:

$$f_\chi(t) = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} t^{\frac{k}{2}-1} e^{-\frac{t}{2}}$$

avec :

$$\Gamma(z) = \int_0^{+\infty} t^{z-1}\, e^{-t}\, \mathrm{d}t$$

et de fonction de répartition $F_\chi(x) = \int_0^x f_\chi(t)dt$. Le seuil χ_{Th} est égal à x tel que $F_\chi(x) = 1 - Err$.

Le test du χ^2 de Pearson est le test de χ^2 couramment utilisé. Cependant, il présente le désavantage de mal se comporter si les $N^*(c)$ prennent des valeurs faibles. Ce test a alors été complété par différentes variantes proposées par Neyman, Freeman et Tukey, Wilks et Kullback :

$$CHI2_{Neyman}(S_{vt}, S_{alg}) = \sum_{c \in \mathcal{Y}} \frac{(N(c) - N^*(c))^2}{N(c)} \tag{2.89}$$

$$CHI2_{Freeman-Tukey}(S_{vt}, S_{alg}) = 4 \sum_{c \in \mathcal{Y}} \left(\sqrt{N(c)} - \sqrt{N^*(c)}\right)^2 \tag{2.90}$$

$$CHI2_{Wilks}(S_{vt}, S_{alg}) = 2 \sum_{c \in \mathcal{Y}} N(c)\ ln(\frac{N(c)}{N^*(c)}) \tag{2.91}$$

$$CHI2_{Kullback}(S_{vt}, S_{alg}) = 2 \sum_{c \in \mathcal{Y}} N^*(c)\ ln(\frac{N^*(c)}{N(c)}) \tag{2.92}$$

2.6.2 Évaluation non supervisée

Il n'y a pas réellement de méthode d'évaluation non supervisée de l'interprétation, c'est à dire sans l'utilisation de connaissances a priori telle qu'une vérité terrain. Cependant, il existe un certain nombre de tests que l'on peut effectuer afin de vérifier la cohérence des résultats obtenus par plusieurs algorithmes différents. Cette évaluation peut également juger de l'adéquation des résultats d'évaluation d'algorithmes avec un ensemble de résultats d'évaluation en provenance d'experts.

Test d'indépendance du χ^2

Le test du χ^2 [66] peut aussi être utilisé afin d'évaluer l'homogénéité des résultats obtenus par plusieurs algorithmes. Il s'agit alors d'estimer les résultats moyens obtenus par les algorithmes et de considérer ces résultats moyens comme étant les résultats souhaités de la part des algorithmes.

Nous disposons donc de plusieurs séries S_{alg} de données correspondant chacune à un résultat d'algorithme :

$$S_{alg} = \{N_{alg}(c) | c \in \mathcal{Y}\}$$

où $N_{alg}(c)$ correspond au nombre d'objets de la classe c reconnus par l'algorithme alg. A partir de ces séries de résultats, on calcule la série de résultats estimés $E_{alg} = \bigcup_{c \in \mathcal{Y}} E_{alg}(c)$ pour chaque algorithme, avec :

$$E_{alg}(c) = \frac{N_+(c) \cdot N_{alg}(+)}{N}$$

avec

- $N^*(\mathcal{Y})$ correspond au nombre de classes,
- $N(alg)$ correspond au nombre d'algorithmes testés,
- $N = \sum_{alg=1}^{N(alg)} \sum_{c \in \mathcal{Y}} N_{alg}(c)$,
- $N_+(c) = \sum_{alg=1}^{N(alg)} N_{alg}(c)$,
- $N_{alg}(+) = \sum_{c \in \mathcal{Y}} N_{alg}(c)$.

On calcule ensuite la valeur du χ^2 ainsi :

$$CHI2_{Pearson}(E_{alg}, S_{alg}) = \sum_{alg=1}^{N(alg)} \sum_{c \in \mathcal{Y}} \frac{(N_{alg}(c) - E_{alg}(c))^2}{E_{alg}(c)} \tag{2.93}$$

Nous considérerons que les algorithmes ont un comportement homogène si la valeur obtenue est inférieure à une valeur seuil χ_{Th} calculée comme précédemment. Le degré de liberté k est alors égal à $(N^*(\mathcal{Y}) - 1) * (N(alg) - 1)$. Il est également possible d'effectuer ce test de χ^2 à partir des variantes proposées par Neyman, Freeman et Tukey, Wilks et Kullback [66].

Mesure de Kappa

La mesure de Kappa [67, 68] est une autre mesure qui permet de rendre compte de l'homogénéité des résultats obtenus par différents algorithmes. Le calcul du coefficient de Kappa K se fait ainsi :

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.94}$$

avec :

- $P(A)$ est la proportion d'algorithmes qui sont d'accord, c'est-à-dire qui classent de manière identique les objets :

$$P(A) = \frac{1}{N(obj)} \sum_{n \in B} S_n \tag{2.95}$$

- $P(E)$ est la proportion *a priori* d'algorithmes en accord, c'est-à-dire la proportion due uniquement au hasard :

$$P(E) = \frac{1}{(N(obj) * N(alg))^2} * \sum_{c \in \mathcal{Y}} T_c^2 \tag{2.96}$$

- $N(obj)$ est le nombre total d'objets dans la base d'images, on a donc $N(obj) = \sum_{j=1}^{N^*(B)} N(I_j)$,
- $N^*(\mathcal{Y})$ correspond au nombre de classes,
- $N(alg)$ correspond au nombre d'algorithmes testés,
- S_i correspond à l'accord entre les algorithmes lors de la classification de l'objet i :

$$S_i = \frac{1}{N(alg) \cdot (N(alg) - 1)} \sum_{c \in \mathcal{Y}} n_i^c * (n_i^c - 1) \tag{2.97}$$

– T_c correspond au nombre d'objets reconnus comme appartenant à la classe c :

$$T_c = \sum_{i=1}^{N(obj)} n_i^c \tag{2.98}$$

– n_i^c correspond au nombre d'algorithmes qui ont reconnu l'objet i comme appartenant à la classe c.

D'après Krippendorff [69], la mesure de Kappa doit être supérieure à 0.8 pour montrer une bonne cohérence entre les algorithmes tandis qu'une valeur inférieure à 0.68 indique que les algorithmes sont globalement en désaccord.

Mesure d'erreur de Bayes

L'erreur de Bayes permet d'estimer le taux d'erreur de reconnaissance. Cette erreur se calcule ainsi :

$$BAY = 1 - \frac{1}{N(obj)} \sum_{i \in B} \max_c \frac{n_i^c}{N(alg)} \tag{2.99}$$

avec n_i^c le nombre d'algorithmes affectant la classe c à l'objet i. Les algorithmes donnent tous le même résultat si l'erreur de Bayes est nulle. L'implémentation de la règle de Bayes afin de mesurer le taux d'erreur revient à considérer que la classe majoritairement reconnue pour l'objet i est la classe réelle de cet objet.

2.7 Conclusions

Si l'évaluation des algorithmes d'interprétation d'images est un challenge important, nous avons vu que pour atteindre cet objectif de nombreux problèmes doivent être surmontés, notamment la multiplicité des formes que prennent les résultats d'interprétation.

Nous pouvons voir qu'il existe un grand nombre de métriques d'évaluation pour la localisation, notamment celles issues des méthodes d'évaluation de la segmentation. Cependant, les objets à évaluer dans le cas de la localisation sont différents de ceux issus de la segmentation. Les performances de ces mesures sont donc à tester dans le cas de la localisation d'objets. Enfin, des métriques proposées pour certaines compétitions (Pascal, Robin...) n'ont pas du tout été étudiées. En effet, ces métriques ont

été définies grâce à l'intuition et l'expérience de chercheurs dans le domaine mais il est difficile de dire *a priori* si elles permettent de prendre en compte tous les cas de figure.

De plus, il reste un certain nombre de problèmes à résoudre si l'on souhaite comparer de façon fiable différentes métriques d'évaluation. D'une part, le passage entre les différentes formes de localisation n'est pas toujours aisé. En effet, comme nous l'avons vu, une frontière peut être définie de plusieurs façons différentes. Il faut également tenir compte, lorsqu'on travaille sur des images réelles, de la confiance accordée aux vérités terrain car celles-ci sont faites par des humains et n'ont donc pas forcément toutes le même degré de précision. D'autre part, de nombreuses métriques font appel à différents paramètres dont il convient d'étudier l'influence.

Nous pouvons également constater que le nombre de méthodes d'évaluation de l'interprétation est assez réduit. Cela provient du fait que l'utilisation des courbes ROC et des courbes de Précision/Rappel est largement répandue. Ce type d'évaluation ne permet pas de quantifier la qualité d'un seul résultat d'interprétation même si l'on connaît la vérité terrain associée.

De plus, nous pouvons voir que le coefficient de confiance renvoyé par les algorithmes n'est pas directement utilisé dans leur évaluation. En effet, on utilise un seuil ou une fonction de décision permettant d'obtenir une décision finale pour chaque objet, perdant ainsi la confiance accordée à chaque résultat.

Nous disposons donc de peu d'informations sur la qualité des métriques d'évaluation des résultats de localisation et de reconnaissance dans le cas particulier de l'interprétation d'images. Le chapitre suivant présente trois études comparatives. La première concerne les métriques de localisation, tandis que les deux suivantes concernent la reconnaissance.

Chapitre 3

Études comparatives

Ce chapitre présente différentes études comparatives de métriques d'évaluation de la localisation et de méthodes de reconnaissance d'objets dans une image. Nous avons tout d'abord formalisé les propriétés que doivent respecter les métriques de localisation. Nous testons sur une base synthétique d'images leur vérification par chaque métrique de la littérature. En ce qui concerne l'évaluation de la reconnaissance, nous présentons une étude comparative sur les modes de représentation d'un objet ainsi qu'une autre sur les descripteurs locaux, comme choix de représentation des objets, pour quantifier la confiance dans une affectation à une classe.

Sommaire

3.1 Introduction

Le chapitre précédent présente les méthodes de l'état de l'art permettant d'évaluer des algorithmes d'interprétation d'images. Cependant, nous ne pouvons en faire ressortir une méthode permettant d'évaluer un résultat d'interprétation. Afin de développer une méthode générique, nous devons donc définir sur quelles métriques

de localisation et de reconnaissance il convient de nous appuyer. Ceci fait l'objet de ce chapitre.

Nous présentons ici trois études comparatives, une concernant la localisation, et deux autres concernant la reconnaissance. À partir de cela, nous allons pouvoir faire ressortir les outils nécessaires pour la création d'une méthode générique d'évaluation de l'interprétation d'images.

3.2 Évaluation des métriques de localisation

Nous avons vu au cours de l'état de l'art qu'il existe un nombre important de métriques permettant d'évaluer un résultat de localisation. Ces métriques ont été principalement développées dans le cadre de l'évaluation de la segmentation. Notre objectif est ici de faire une étude permettant de comparer les performances des différentes métriques existantes dans le cadre particulier de l'évaluation de la localisation afin de mettre en avant la métrique la plus efficace pour évaluer correctement un résultat de localisation.

Afin de choisir une métrique, nous devons nous intéresser à certaines propriétés nous permettant de les différencier. Les propriétés recherchées pourront être, par exemple, la monotonie ou la continuité, la vérification de notions de symétrie...

3.2.1 Protocole

Notre étude s'appuie sur le principe de l'évaluation supervisée de la localisation : on dispose d'une vérité terrain et d'un résultat de localisation, la métrique d'évaluation nous permet de donner une note de pertinence entre ces deux données. Afin d'étudier une métrique, nous avons remplacé la vérité terrain et le résultat de localisation par deux images synthétiques : une des images nous sert de vérité terrain tandis que l'autre correspond à un résultat simulé de localisation. Le résultat de localisation utilisé est obtenu en altérant de façon contrôlée la vérité terrain. Nous étudions ensuite l'évolution de la métrique en fonction de l'altération que l'on contrôle (voir Fig. 3.1). Cela nous permet de vérifier le bon comportement d'une métrique en fonction d'un type particulier d'altération.

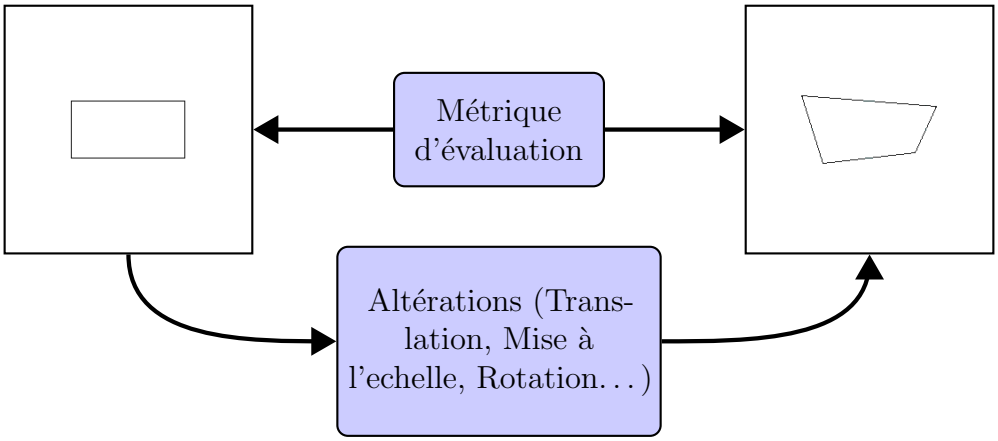


FIGURE 3.1 – Principe de l'étude comparative

Création de la base de vérités terrain

Les vérités terrain que nous utilisons doivent représenter différents cas que nous pouvons rencontrer dans les cas réels d'utilisation d'un algorithme de localisation. Nous avons créé 16 vérités terrain qui vont nous permettre d'étudier les métriques d'évaluation, elles sont présentées à la figure 3.2. Ces vérités terrain reproduisent des résultats de localisation avec des boîtes englobantes de différentes tailles et formes, se situant à proximité ou loin du bord de l'image. Enfin, nous avons pris des vérités terrain représentant des objets réels issus de divers domaines où la localisation est utilisée.

Création de la base de données synthétiques

Les altérations que nous avons appliquées à chaque vérité terrain sont les suivantes : translation, mise à l'échelle, rotation et perspective.

Pour toutes les vérités terrain, nous avons effectué une translation entre -24 et +24 pixels sur chacun des axes. Cela fait 2400 images altérées par vérité terrain. Nous pouvons voir à la figure 3.3 une translation appliquée sur deux vérités terrain, la 5 et la 14, avec trois jeux de paramètres différents : 12 pixels en X ; 24 pixels en Y puis simultanément -12 en X et -24 en Y. Concernant la mise à l'échelle, nous l'avons effectuée sur toutes les vérités terrains avec des paramètres allant de -24 à +24 pixels sur chacun des axes. Nous avons alors également obtenus 2400 images. Les images présentées sur la figure 3.4 sont les résultats de la mise à l'échelle pour les deux vérités terrain 5 et 14 avec les mêmes jeux de paramètres que pour la translation. La rotation ne dépend que d'un paramètre qui est l'angle de rotation. Ce paramètre

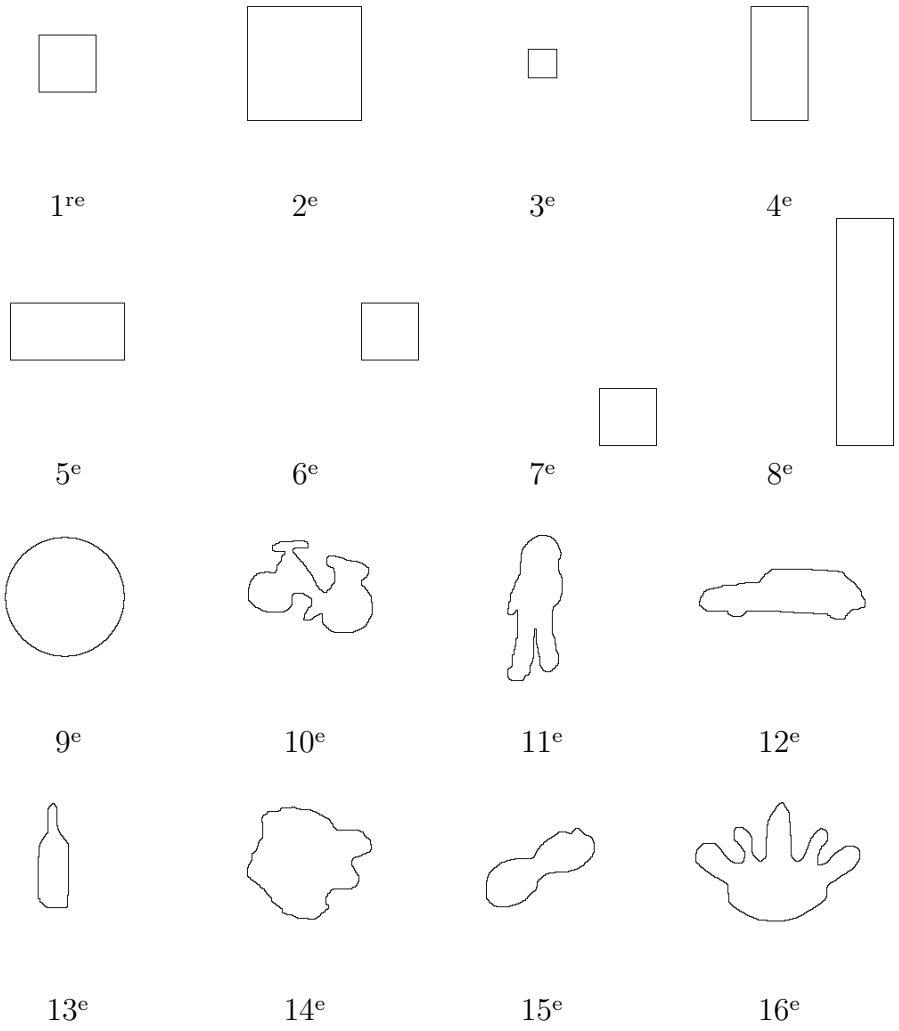


FIGURE 3.2 – Vérités terrain utilisées pour l'étude comparative

varie entre -90 et +90, ce qui fait 180 images altérées par vérité terrain. Les images de la figure 3.5 présentent le résultat d'une rotation sur les vérités terrain 5 et 14 avec un angle de 20, 40 et -40 degrés. La déformation en perspective dépend quant à elle de deux variables jouant sur la perspective sur chacun des axes et allant de -24 à +24 pour chacun (voir Fig. 3.6).

En tout, nous obtenons *7 380* images altérées par vérité terrain, soit un total de *118 080* images altérées.

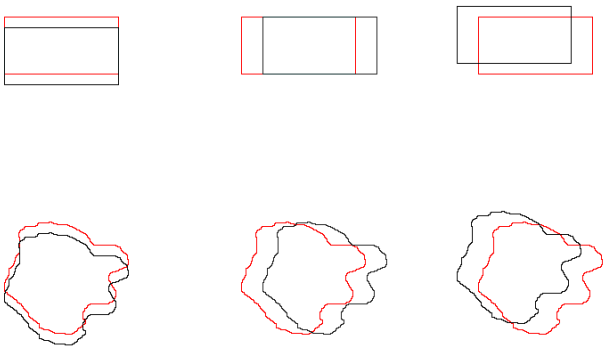


FIGURE 3.3 – Translation pour deux vérités terrain (vérité terrain en rouge)

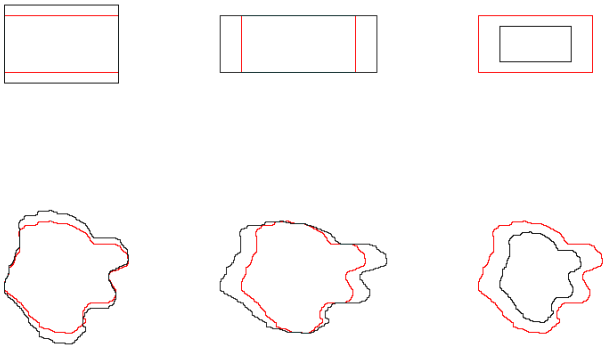


FIGURE 3.4 – Mise à l'échelle pour deux vérités terrain (vérité terrain en rouge)

Propriétés

Les métriques d'évaluation de la localisation sont généralement développées en faisant appel au bon sens et à l'intuition des chercheurs. Cependant, ces métriques, proposées de façon empirique, ne respectent pas forcément un certain nombre de propriétés.

Pour qu'une métrique soit pertinente, il faut qu'elle réagisse correctement à un maximum de propriétés. Nous avons tout d'abord considéré les propriétés des distances, c'est-à-dire, pour une métrique que l'on notera M :

Symétrie : $M(I_1, I_2) = M(I_2, I_1)$, une métrique doit pouvoir évaluer un résultat d'interprétation sans savoir, a priori, si I_1 ou I_2 est la vérité terrain,

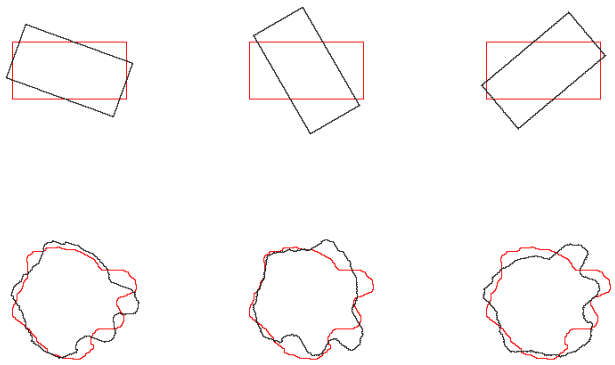


FIGURE 3.5 – Rotation pour deux vérités terrain (vérité terrain en rouge)

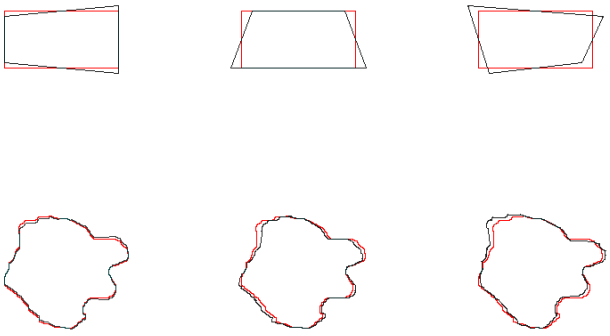


FIGURE 3.6 – Déformation trapézoïdale pour deux vérités terrain (vérité terrain en rouge)

Séparabilité : $M(I_1, I_2) = 0 \Leftrightarrow I_1 = I_2$, une métrique doit différencier un résultat de localisation altéré de la vérité terrain,

Inégalité triangulaire : $M(I_1, I_3) \leqslant M(I_1, I_2) + M(I_2, I_3)$

A ces propriétés, nous avons ajouté les propriétés suivantes :

Symétrie d'axe Oz : une métrique doit pénaliser identiquement une altération dans une direction et dans la direction opposée,

Stricte Monotonie : plus on altère un résultat de localisation par rapport à la vérité terrain, plus la métrique d'évaluation doit pénaliser cette altération,

Régularité uniforme : étant dans un contexte discret, on ne peut à proprement

parler de continuité. On parlera donc de régularité uniforme en vérifiant le fait qu'il n'y ait pas de saut brusque (décrochage) dans la métrique d'évaluation,

Dépendance topologique : la métrique doit prendre en compte la forme et la taille des objets localisés.

Nous allons donc vérifier si certaines propriétés attendues sont vérifiées grâce aux résultats de notre étude d'évaluation des différentes métriques, qui prennent la forme de courbes en 3 dimensions. On peut voir sur la figure 3.7 un exemple de courbes obtenues pour trois métriques différentes en considérant une altération et une vérité terrain.

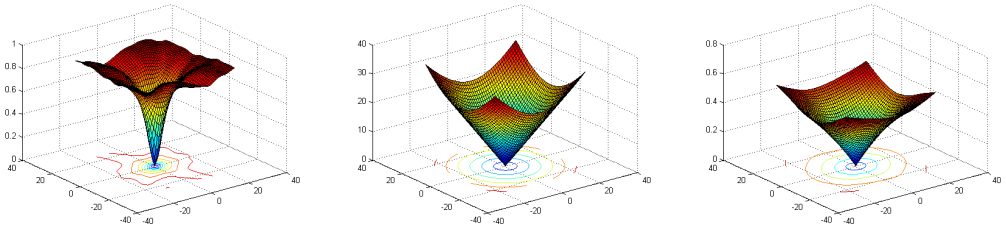


FIGURE 3.7 – Exemple de résultats d'évaluation pour une vérité terrain, une altération et trois métriques d'évaluation

Soit $M(I_{vt}, I_{alt\_X\_Y})$ le résultat de la métrique M pour l'altération alt de paramètres X et Y. Pour vérifier si une métrique est symétrique, nous vérifions que les résultats sont symétriques par rapport à l'axe OZ, c'est-à-dire que le résultat est identique dans un sens et dans le sens opposé : $M(I_{vt}, I_{alt\_X\_Y}) = M(I_{vt}, I_{alt\_-X\_-Y})$. On souhaite un résultat symétrique pour la translation, la rotation et la perspective. Dans le cas de la rotation, nous vérifions que $M(I_{vt}, I_{alt\_D}) = M(I_{vt}, I_{alt\_-D})$

Afin de s'assurer que la métrique est monotone, nous nous intéressons à la partie des résultats où X et Y sont positifs. On pourra également s'intéresser au cas où X et Y peuvent être négatifs en adaptant les critères. Nous vérifions que plus on altère le résultat de localisation, plus la métrique est pénalisante, c'est-à-dire si $M(I_{vt}, I_{alt\_X\_Y}) \leqslant M(I_{vt}, I_{alt\_X+1\_Y})$, $M(I_{vt}, I_{alt\_X\_Y}) \leqslant M(I_{vt}, I_{alt\_X\_Y+1})$ et $M(I_{vt}, I_{alt\_X\_Y}) \leqslant M(I_{vt}, I_{alt\_X+1\_Y+1})$. Dans le cas de la rotation, on attend simplement que $M(I_{vt}, I_{alt\_D}) \leqslant M(I_{vt}, I_{alt\_D+1})$. Si l'on souhaite vérifier si une métrique est strictement monotone, il faut alors utiliser des inégalités strictes. On attend un résultat strictement monotone pour la translation, la rotation et la perspective. On attend une stricte monotonie pour la mise à l'échelle dans le cas où X et Y sont tous

les deux soit positifs soit négatifs.

Pour la régularité, nous nous intéressons également à la partie des résultats où X et Y sont positifs. Nous vérifions que la différence entre deux résultats successifs $M(I_{vt}, I_{alt\_X\_Y})$ et $M(I_{vt}, I_{alt\_X+1\_Y})$ n'est pas trop importante. Pour cela, nous avons fixé empiriquement une valeur du seuil T, égale à un huitième de l'amplitude de la métrique, que $|M(I_{vt}, I_{alt\_X\_Y}) - M(I_{vt}, I_{alt\_X+1\_Y})|$ ne doit pas dépasser. Si tel est le cas, alors soit la métrique a une forte pente, soit il y a une rupture de régularité. Pour vérifier s'il y a cette rupture, nous comparons alors la différence $|M(I_{vt}, I_{alt\_X\_Y}) - M(I_{vt}, I_{alt\_X+1\_Y})|$ avec la précédente $|M(I_{vt}, I_{alt\_X-1\_Y}) - M(I_{vt}, I_{alt\_X\_Y})|$ et la suivante $|M(I_{vt}, I_{alt\_X+1\_Y}) - M(I_{vt}, I_{alt\_X+2\_Y})|$. Si la différence $|M(I_{vt}, I_{alt\_X\_Y}) - M(I_{vt}, I_{alt\_X+1\_Y})|$ est 4 fois supérieure aux autres, alors, on considère que la métrique n'est pas régulière. On effectue de même sur la coordonnée Y. On vérifie qu'une métrique est régulière pour la translation, la mise à l'échelle, la rotation et la perspective.

Enfin, nous regardons si le résultat de la métrique dépend de la forme et de la taille de l'objet localisé. Afin de vérifier si la forme de l'objet joue un rôle, nous comparons les résultats de localisation sur deux vérités terrains différentes : la 4 et la 5. Sur ces deux vérités terrain, l'objet localisé est un rectangle, en hauteur pour la vérité terrain 4 et en largeur pour la vérité terrain 5. Ainsi, si les résultats sont les mêmes pour une métrique pour une même altération et ces deux vérités terrain, c'est que les résultats sont indépendants de la forme de l'objet. On considère qu'une bonne métrique doit prendre en compte la forme de l'objet. On attend d'une métrique qu'elle pénalise différemment selon la taille de l'objet. Pour cela, on va comparer les résultats des vérités terrain 1 et 2, représentant toutes les deux un carré de taille différente, plus grand sur la seconde vérité terrain. Deux comportements peuvent être recherchés. On peut considérer que plus l'objet est petit, moins il doit être pris en compte dans l'image. Les résultats liés à la vérité terrain 1 devront dans ce cas être inférieurs à ceux de la vérité terrain 2. On peut également considérer qu'une altération de même ampleur a plus d'influence sur un petit objet et ce seront alors les résultats inverses qui seront recherchés.

La méthode que nous avons mise en place pour étudier les performances d'une métrique d'évaluation de la localisation consiste à vérifier si elle remplit les différentes propriétés attendues pour toutes les altérations considérées.

Les métriques étudiées

Les métriques comparées proviennent de l'état de l'art du chapitre précèdent. La liste des métriques étudiées est présentée dans le tableau 3.1.

TABLE 3.1: Liste des métriques utilisées au cours de l'étude comparative

Métrique	Type	Référence
ROB_{loc}	Boite	[2]
ROB_{com}	Boite	[2]
ROB_{cor}	Boite	[2]
$ErrLoc$	Contour	[33, 34]
$ErrSous$	Contour	[33, 34]
$ErrSur$	Contour	[33, 34]
SNR	Contour	[35, 36]
RMS	Contour	[35, 36]
Lq	Contour	[35, 36]
KUL	Contour	[33, 37]
BAH	Contour	[33, 37]
JEN	Contour	[33, 37]
$DMoy$	Contour	[33, 39]
$DMoC$	Contour	[33, 39]
FOM	Contour	[39, 40]
HAU	Contour	[34, 41]
BAD	Contour	[34, 36]

Métrique	Type	Référence
ODI_n	Contour	[33, 42]
UDI_n	Contour	[33, 42]
PAS	Masque	[43]
$HEN1$	Masque	[44]
$HEN2$	Masque	[44]
$YAS1$	Masque	[33, 45]
$YAS2$	Masque	[33, 45]
$YAS3$	Masque	[33, 45]
$MAR1$	Masque	[33, 47]
$MAR2$	Masque	[33, 47]
HAM	Masque	[33, 48]
$HAF1$	Masque	[49]
$HAF2$	Masque	[50]
VIN	Masque	[51, 52]
Ppx	Masque	[46]
Rpx	Masque	[46]

Pour la métrique Lq, deux valeurs du paramètre q ont été testées : q=1 et q=3. La distance de Baddeley BAD a été testée avec le paramètre $p = 1$, $p = 2$ et $p = 3$. Enfin, les métriques de Odet ODI_n et UDI_n ont été testées avec la distance seuil $d_{TH} = 5$ et le paramètre p =1 et $p = 2$. Enfin, nous avons modifié certaines métriques afin que le résultat optimal soit 0 pour toutes, et que, lorsqu'on altère la vérité terrain, le résultat de la métrique soit croissant.

Cela nous fait donc en tout 38 métriques à calculer pour toutes les altérations et vérités terrain vues précédemment donnant *3 040* résultats à étudier.

3.2.2 Résultats

Distance

Le tableau 3.2 présente les résultats obtenus pour les propriétés de distance. Ce tableau nous permet de voir quelles métriques ne sont pas à proprement parler des

distances car ne respectant pas l'ensemble des propriétés requises.

TABLE 3.2: Résultats pour les propriétés de distance

Métrique	Représentation	Symétrie	Séparabilité	Inégalité triangulaire	Score
ROB_{loc}	Boîte				
ROB_{com}	Boîte	✓			*
ROB_{cor}	Boîte	✓			*
$ErrLoc$	Contour	✓	✓	✓	***
$ErrSous$	Contour		✓	✓	**
$ErrSur$	Contour		✓	✓	**
SNR	Contour				
RMS	Contour	✓	✓	✓	***
$Lq, 1$	Contour	✓	✓	✓	***
$Lq, 3$	Contour	✓	✓	✓	***
KUL	Contour		✓	✓	**
BAH	Contour	✓			*
JEN	Contour		✓	✓	**
$DMoy$	Contour		✓	✓	**
$DMoC$	Contour		✓	✓	**
FOM	Contour		✓	✓	**
HAU	Contour	✓	✓	✓	***
$BAD, 1$	Contour	✓	✓	✓	***
$BAD, 2$	Contour	✓	✓	✓	***
$BAD, 3$	Contour	✓	✓	✓	***
$ODI_n, 1$	Contour		✓	✓	**
$ODI_n, 2$	Contour		✓	✓	**
$UDI_n, 1$	Contour		✓	✓	**
$UDI_n, 2$	Contour		✓	✓	**
PAS	Masque	✓	✓	✓	***
$HEN1$	Masque				
$HEN2$	Masque		✓	✓	**
$YAS1$	Masque				
$YAS2$	Masque				
$YAS3$	Masque				
MAR_{gce}	Masque	✓	✓	✓	***
MAR_{lce}	Masque	✓	✓	✓	***
HAM	Masque	✓	✓	✓	***
$HAF1$	Masque	✓	✓	✓	***
$HAF2$	Masque		✓	✓	**
VIN	Masque	✓	✓	✓	***
P_{px}	Masque				
R_{px}	Masque				

Nous pouvons voir que la majorité des métriques étudiées ne satisfont pas les trois propriétés, en particulier la symétrie. La propriété la plus intéressante est la

séparabilité, car elle permet de mettre en évidence le fait que certaines métriques sont incapables de pénaliser certaines altérations. Il est intéressant de constater qu'une métrique remplissant la propriété de séparabilité satisfait également à l'inégalité triangulaire.

Translation

La table 3.3 présente les résultats obtenus pour l'altération de translation. Nous constatons que les métriques obtiennent de bons résultats dans l'ensemble, avec au moins trois propriétés satisfaites sur les cinq. Nous pouvons tout d'abord voir que les résultats sont symétriques pour toutes les métriques, ce qui indique que les métriques utilisées pénalisent identiquement une translation de 10 pixels vers le haut ou vers le bas de l'image par exemple. Cependant, trois métriques sont insensibles à une altération de type translation : ROB_{cor}, ROB_{com} et $HEN1$. Ces trois métriques renvoient un résultat nul quelque soit la translation appliquée à la vérité terrain. Les trois métriques de la compétition Robin évoluent en fait comme on pouvait s'y attendre : ROB_{loc} pénalise correctement la translation tandis que ROB_{cor} et ROB_{com} ne pénalisent pas ce type d'altération. Nous pouvons également constater que les métriques basées contour obtiennent, exceptées $DMoy$, $DMoC$, FOM et HAU, de moins bons résultats que les métriques basées masque, principalement car elles ne satisfont pas à la régularité uniforme et la stricte monotonie.

Concernant la propriété de dépendance de taille, nous pouvons voir à la figure 3.8, les trois différents cas où les résultats de l'évaluation dépendent de la taille de l'objet localisé. L'altération considérée est une translation sur l'axe vertical, avec une altération constante sur l'axe horizontal de 24 pixels. La courbe pleine correspond au résultat de l'évaluation de la 1re vérité terrain, et la courbe en pointillé à l'évaluation de la 2e vérité terrain qui est la plus grande. Nous pouvons voir que la métrique $DMoy$ pénalise moins une altération sur un petit objet, ce que nous notons ✓−, alors que la métrique $HAF1$ pénalise plus une altération sur un petit objet, ce que nous notons ✓+. Nous pouvons également remarquer que la non-régularité de la métrique $HAF1$ apparaît uniquement avec le plus petit objet. La métrique FOM, quant à elle, pénalise plus les petits objets pour une petite altération, et pénalise plus un plus grand objet pour une plus grande altération, ce que nous notons ✓ + /−.

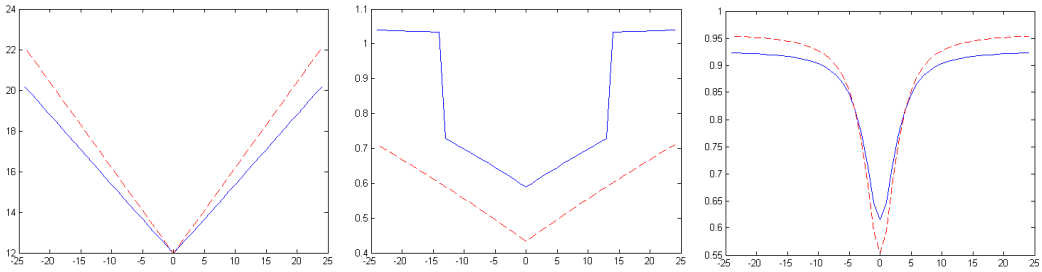
On peut voir que $ErrLoc$, $ErrSous$, $ErrSur$, SNR, RMS et les distances L_q, de Bhattacharyya (BAH), de Küllback (KUL) et de Jensen (JEN) donnent de mauvais résultats. En effet, ces métriques pénalisent très fortement et indifféremment une

TABLE 3.3: Résultats pour l'altération de translation

Métrique	Représentation	Symétrie d'axe Oz	Stricte monotonie	Régularité uniforme	Dépendance de taille	Dépendance de forme	Score
ROB_{loc}	Boîte	✓	✓	✓	✓+	✓	*****
ROB_{com}	Boîte						
ROB_{cor}	Boîte						
$ErrLoc$	Contour	✓			✓-	✓	***
$ErrSous$	Contour	✓			✓+/-	✓	**
$ErrSur$	Contour	✓			✓-	✓	***
SNR	Contour	✓			✓+	✓	***
RMS	Contour	✓			✓-	✓	***
$Lq, 1$	Contour	✓			✓-	✓	***
$Lq, 3$	Contour	✓			✓-	✓	***
KUL	Contour	✓			✓-	✓	***
BAH	Contour	✓			✓-	✓	***
JEN	Contour	✓			✓-	✓	***
$DMoy$	Contour	✓	✓	✓	✓-	✓	*****
$DMoC$	Contour	✓	✓	✓	✓-	✓	*****
FOM	Contour	✓	✓	✓	✓+/-	✓	****
HAU	Contour	✓	✓	✓			***
$BAD, 1$	Contour	✓		✓	✓-	✓	****
$BAD, 2$	Contour	✓			✓+	✓	***
$BAD, 3$	Contour	✓			✓+	✓	***
$ODI_n, 1$	Contour	✓			✓+/-	✓	**
$ODI_n, 2$	Contour	✓			✓-	✓	***
$UDI_n, 1$	Contour	✓			✓+/-	✓	**
$UDI_n, 2$	Contour	✓			✓-	✓	***
PAS	Masque	✓	✓	✓	✓+	✓	*****
$HEN1$	Masque						
$HEN2$	Masque	✓	✓	✓	✓+	✓	*****
$YAS1$	Masque	✓	✓	✓	✓+	✓	*****
$YAS2$	Masque	✓	✓	✓	✓-	✓	*****
$YAS3$	Masque	✓	✓	✓	✓-	✓	*****
MAR_{gce}	Masque	✓	✓	✓	✓-	✓	*****
MAR_{lce}	Masque	✓	✓	✓	✓-	✓	*****
HAM	Masque	✓		✓	✓-	✓	****
$HAF1$	Masque	✓	✓		✓+	✓	****
$HAF2$	Masque	✓	✓		✓-	✓	****
VIN	Masque	✓	✓	✓	✓-	✓	*****
P_{px}	Masque	✓	✓	✓	✓+	✓	*****
R_{px}	Masque	✓	✓	✓	✓+	✓	*****

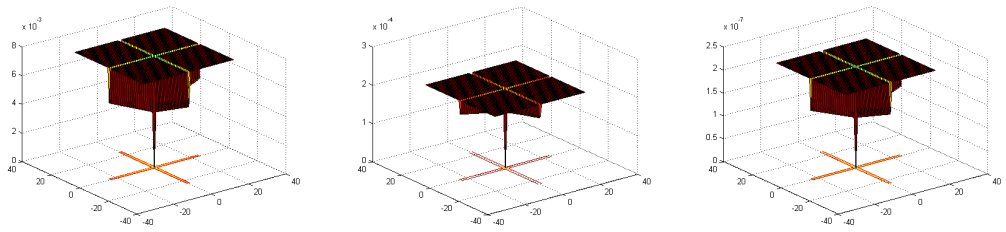
translation d'un seul ou de 16 pixels comme on peut le voir sur la figure 3.9.

La distance moyenne ($DMoy$), son carré ($DMoC$) et la métrique de Pratt (FOM) évoluent correctement, bien qu'ayant des comportements totalement différents. On peut voir sur la figure 3.10 les résultats de ces métriques ainsi qu'une coupe selon l'axe vertical. Nous pouvons voir que la distance moyenne évolue linéairement par rapport à la translation : une translation de 10 pixels obtient une note deux fois



(a) $DMoy$ (b) $HAF1$ (c) FOM

FIGURE 3.8 – Exemples d'évaluation de métrique de localisation : différents comportement concernant la propriété de dépendance de taille (la courbe pleine correspond au résultat d'évaluation de la plus petite vérité terrain, la courbe en pointillé à l'évaluation de la plus grande)

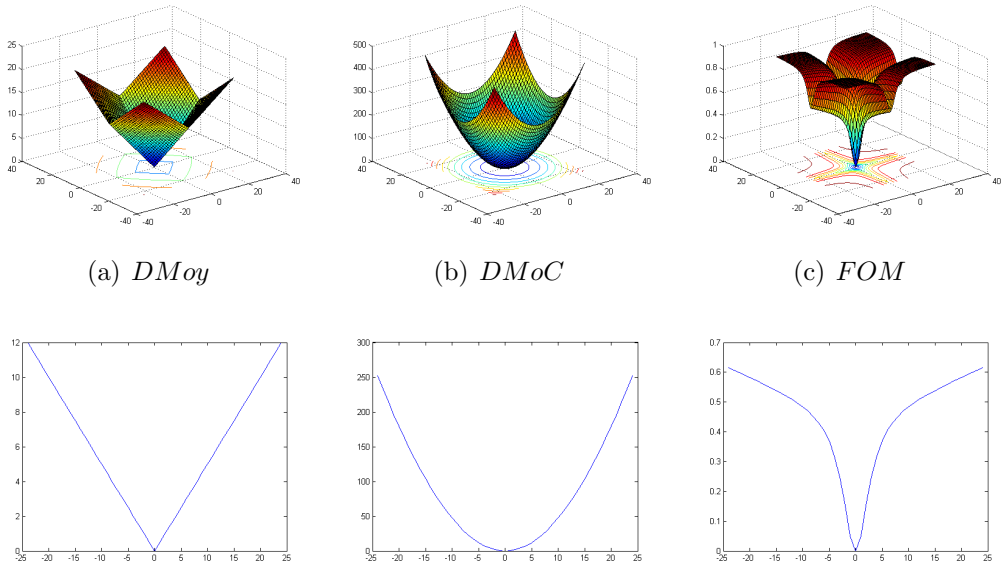


(a) $ErrLoc$ (b) RMS (c) JEN

FIGURE 3.9 – Résultats des métriques $ErrLoc$, RMS et JEN pour une translation sur la première vérité terrain.

moins bonne qu'une translation de 5 pixels. Le carré de la distance moyenne va lui être peu pénalisant pour les petites translations puis va pénaliser de plus en plus fortement les grandes translations. La métrique de Pratt est quant à elle assez pénalisante dès de petites translations.

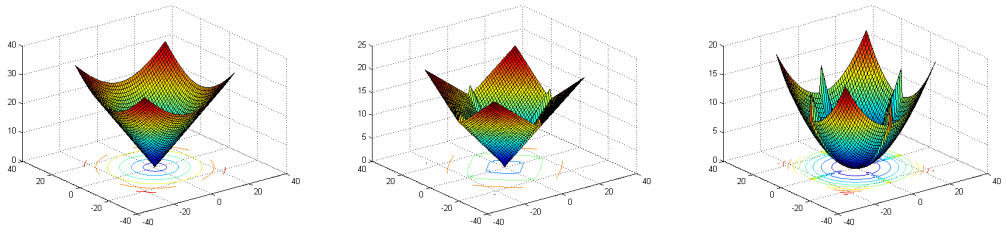
La distance de Hausdorff (HAU) évolue correctement, mais ne tient cependant pas compte de la taille et de la forme de l'objet évalué. A l'inverse, la distance de Baddeley ($BAD, 1$) en tient compte, mais a cependant un défaut que l'on peut voir sur la figure 3.11 : lorsque l'altération n'a lieu que sur un seul axe, celle-ci est plus fortement pénalisée, ce qui crée un décrochage. Les métriques de Odet (ODI_n, UDI_n) ont le même problème. Cependant, l'avantage de ces deux métriques est d'être paramétrables et donc de pouvoir présenter des comportements différents. Lorsque le paramètre p vaut 1, le comportement des métriques est linéaire, comme la distance moyenne, quand p vaut 2, le comportement est alors identique au carré de la distance moyenne (comme on pouvait s'y attendre).



(a) $DMoy$ (b) $DMoC$ (c) FOM

(d) $DMoy$ (e) $DMoC$ (f) FOM

FIGURE 3.10 – Résultats des métriques $DMoy$, $DMoC$ et FOM pour une translation sur la première vérité terrain.

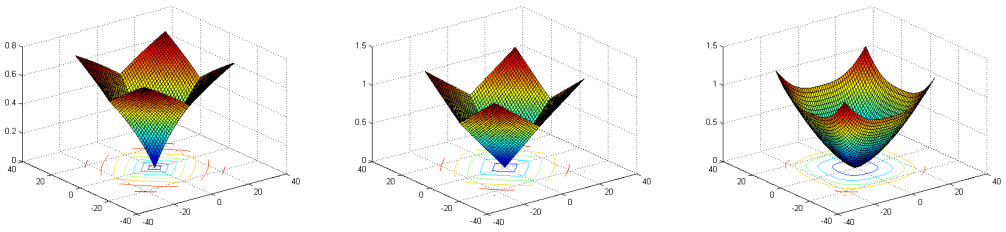


(a) HAU (b) $BAD, 1$ (c) $ODI_n, 2$

FIGURE 3.11 – Résultats des métriques HAU, $BAD, 1$ et $ODI_n, 2$ pour une translation sur la première vérité terrain.

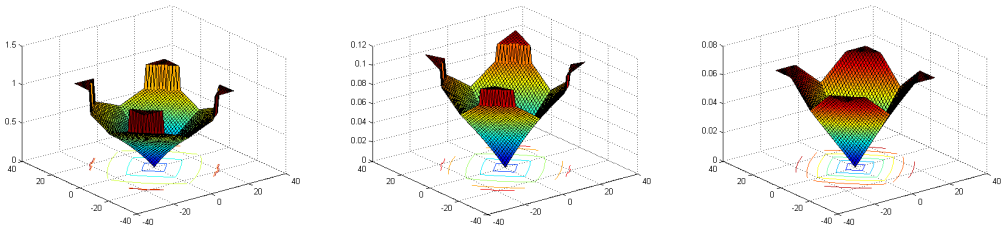
Les métriques basées sur des masques donnent en général de meilleurs résultats. Les métriques du Pascal VOC challenge (PAS), de Henricsson ($HEN1, HEN2$), de Yasnoff ($YAS1, YAS2, YAS3$), de Martin (MAR_{gce}, MAR_{lce}) et de Vinet (VIN) donnent de très bons résultats. Mis à part $YAS3$, ces métriques évoluent linéairement comme on peut le voir sur la figure 3.12.

Enfin, les métriques de Hafiane et la distance de Hamming ont des notes légèrement moins bonnes : les métriques de Hafiane présentent une discontinuité et la distance de Hamming n'est pas strictement croissante (voir Fig. 3.13). Ces métriques ont des problèmes lorsque la translation est trop importante. Lors de la mise en correspondance, l'objet translaté n'est plus mis en correspondance avec l'objet de la vérité terrain, mais avec le fond, ce qui crée ce problème.



(a) PAS (b) $HEN2$ (c) $YAS3$

FIGURE 3.12 – Résultats des métriques PAS, $HEN2$ et $YAS3$ pour une translation sur la première vérité terrain.



(a) $HAF1$ (b) $HAF2$ (c) HAM

FIGURE 3.13 – Résultats des métriques $HAF1$, $HAF2$ et HAM pour une translation sur la première vérité terrain.

Changement d'échelle

Nous pouvons voir à la table 3.4, les résultats obtenus pour l'altération de changement d'échelle.

Les trois métriques du projet Robin évoluent comme attendu : ROB_{loc} ne pénalise pas du tout les changements d'échelle, alors que ROB_{cor} et ROB_{com} les pénalisent. Nous pouvons noter un mauvais comportement de ROB_{com} lorsque X est positif et Y négatif, et réciproquement. En effet, lorsque $X = -Y$, cette métrique renvoie 0. ROB_{cor} a un mauvais comportement lorsque X et Y sont tous les deux positifs ou tous les deux négatifs. En particulier, cette métrique renvoie 0 lorsque $X = Y$.

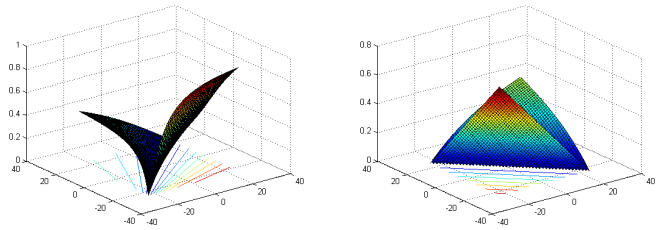
Les métriques basées contour, $ErrLoc$, $ErrSous$, $ErrSur$, SNR, RMS et les distances L_q, Bhattacharyya (BAH), Küllback (KUL) et Jensen (JEN), obtiennent de mauvais résultats en pénalisant fortement de petites altérations (voir Fig. 3.15). De plus, on peut voir un mauvais comportement quand il y a une réduction (X et Y négatifs). En effet, plus l'objet est altéré et moins ces métriques le pénalisent.

TABLE 3.4: Résultats pour l'altération de changement d'échelle

Métrique	Représentation	Stricte monotonie	Régularité uniforme	Dépendance de taille	Dépendance de forme	Score
ROB_{loc}	Boîte					
ROB_{com}	Boîte	✓	✓	✓+		***
ROB_{cor}	Boîte		✓	✓+	✓	***
$ErrLoc$	Contour			✓-	✓	**
$ErrSous$	Contour			✓+/-	✓	*
$ErrSur$	Contour			✓-	✓	**
SNR	Contour			✓+	✓	**
RMS	Contour			✓-	✓	**
$Lq, 1$	Contour			✓-	✓	**
$Lq, 3$	Contour			✓-	✓	**
KUL	Contour			✓-	✓	**
BAH	Contour	✓	✓	✓-	✓	****
JEN	Contour			✓-	✓	**
$DMoy$	Contour	✓	✓	✓+/-	✓	***
$DMoC$	Contour	✓	✓	✓+/-	✓	***
FOM	Contour	✓	✓	✓+/-	✓	***
HAU	Contour	✓	✓			**
$BAD, 1$	Contour		✓	✓+/-	✓	**
$BAD, 2$	Contour		✓	✓+	✓	***
$BAD, 3$	Contour		✓	✓+	✓	***
$ODI_n, 1$	Contour			✓+/-	✓	*
$ODI_n, 2$	Contour			✓+/-	✓	*
$UDI_n, 1$	Contour			✓+/-	✓	*
$UDI_n, 2$	Contour			✓+/-	✓	*
PAS	Masque	✓	✓	✓+	✓	****
$HEN1$	Masque	✓	✓	✓+/-	✓	***
$HEN2$	Masque	✓	✓	✓+	✓	****
$YAS1$	Masque		✓	✓+	✓	***
$YAS2$	Masque		✓	✓-	✓	***
$YAS3$	Masque		✓	✓-	✓	***
MAR_{gce}	Masque	✓	✓	✓-	✓	****
MAR_{lce}	Masque	✓	✓	✓-	✓	****
HAM	Masque		✓	✓-	✓	***
$HAF1$	Masque			✓+/-	✓	*
$HAF2$	Masque	✓		✓-	✓	***
VIN	Masque	✓	✓	✓-	✓	****
P_{px}	Masque		✓	✓+	✓	***
R_{px}	Masque		✓	✓+	✓	***

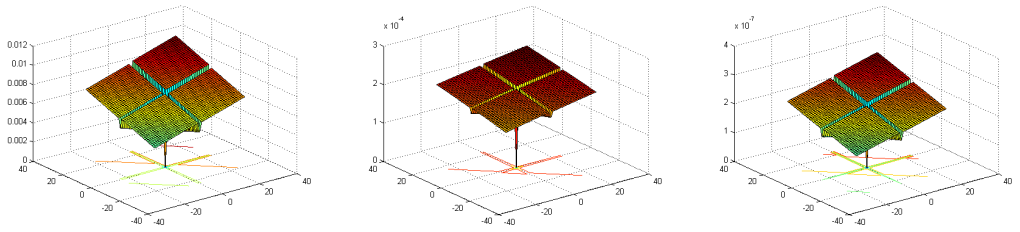
Les métriques $DMoy$, $DMoC$ et FOM ont encore un comportement correct vis-à-vis de la mise à l'échelle comme on peut le voir sur la figure 3.16. On notera toutefois que lorsque X et Y ont des signes opposés, alors l'altération est moins pénalisée que lorsque X et Y sont de même signe.

La métrique de Hausdorff (HAU) donne ici aussi des résultats corrects (voir Fig. 3.17), mais sans tenir compte ni de la taille, ni de la forme de l'objet. La distance



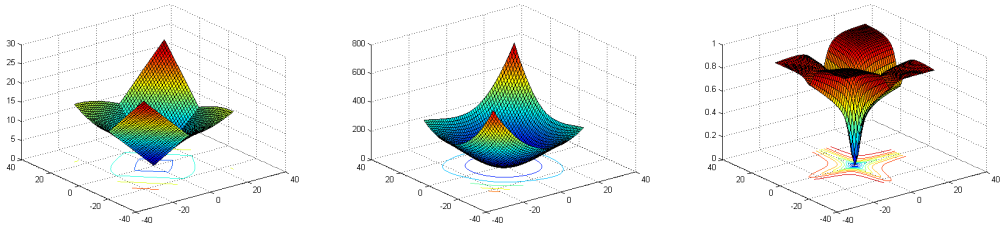
(a) ROB_{com} (b) ROB_{cor}

FIGURE 3.14 – Résultats des métriques ROB_{com} et ROB_{cor} pour une mise à l'échelle sur la première vérité terrain.



(a) $ErrLoc$ (b) RMS (c) JEN

FIGURE 3.15 – Résultats des métriques $ErrLoc$, RMS et JEN pour une mise à l'échelle sur la première vérité terrain.

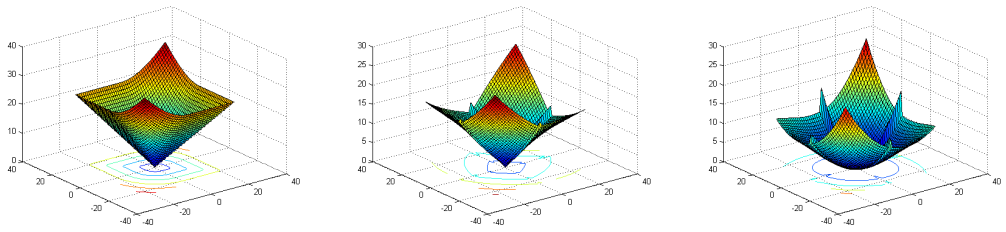


(a) $DMoy$ (b) $DMoC$ (c) FOM

FIGURE 3.16 – Résultats des métriques $DMoy$, $DMoC$ et FOM pour une mise à l'échelle sur la première vérité terrain.

de Baddeley et les métriques d'Odet donnent également des résultats corrects mis à part le problème du décrochage identique à la translation. De plus, on notera que la distance de Baddeley (BAD) et la métrique UDI_n pénalisent plus une réduction et que la métrique ODI_n pénalise plus un agrandissement.

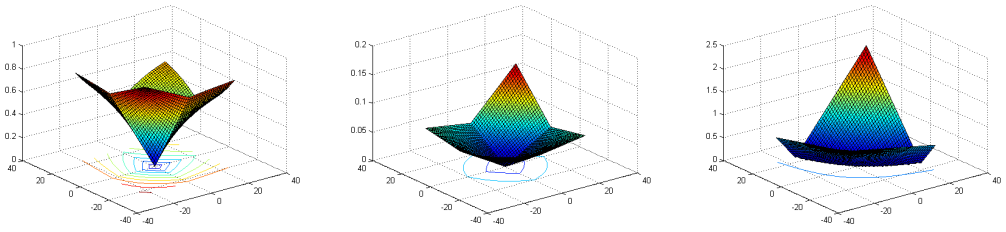
Les métriques basées sur des masques obtiennent des résultats globalement meilleurs, mais pas aussi satisfaisants que dans le cas de la translation. Les métriques



(a) HAU (b) $BAD, 1$ (c) $ODI_n, 2$

FIGURE 3.17 – Résultats des métriques HAU, $BAD, 1$ et $ODI_n, 2$ pour une mise à l'échelle sur la première vérité terrain.

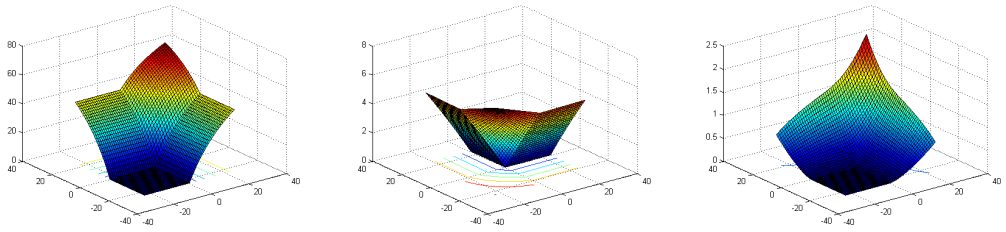
PAS, $HEN2$, MAR_{gce} et VIN obtiennent des résultats satisfaisants (voir Fig. 3.18). On notera que PAS pénalise plus la réduction, et que $HEN2$, MAR_{gce} et VIN pénalisent davantage l'agrandissement. Enfin, la métrique $HEN1$ ne donne pas des résultats corrects.



(a) PAS (b) MAR_{gce} (c) $HEN2$

FIGURE 3.18 – Résultats des métriques PAS, MAR_{gce} et $HEN2$ pour une mise à l'échelle sur la première vérité terrain.

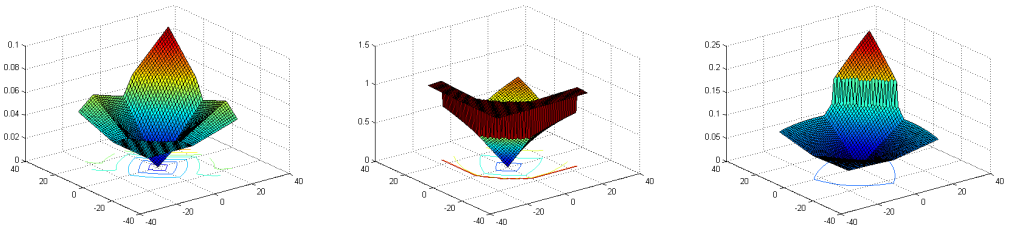
Les métriques HAM, $YAS1$, $YAS2$ et $YAS3$ ont le défaut de ne pas être sensibles soit à l'agrandissement, soit à la réduction, comme on peut le voir à la figure 3.19.



(a) $YAS1$ (b) $YAS2$ (c) $YAS3$

FIGURE 3.19 – Résultats des métriques $YAS1$, $YAS2$ et $YAS3$ pour une mise à l'échelle sur la première vérité terrain.

Enfin, les métriques de Hafiane présentent le même problème de discontinuité due à une mauvaise mise en correspondance, comme on peut le voir à la figure 3.20.



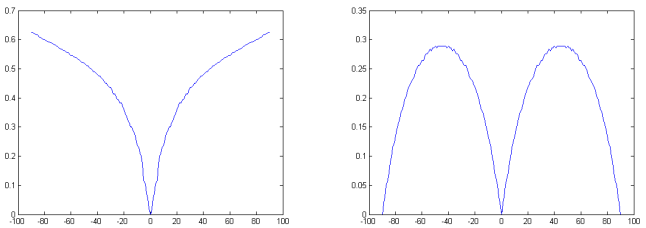
(a) HAM (b) $HAF1$ (c) $HAF2$

FIGURE 3.20 – Résultats des métriques HAM, $HAF1$ et $HAF2$ pour une mise à l'échelle sur la première vérité terrain.

Rotation

La table 3.5 présente les résultats pour une altération de rotation. Nous pouvons voir que toutes les métriques sont symétriques, c'est-à-dire qu'elles pénalisent toutes autant une rotation de D degrés qu'une rotation de $-D$ degrés. On remarque également que les métriques basées masque obtiennent de meilleures performances que les métriques basées contour.

Nous pouvons voir que les métriques du projet Robin réagissent toutes différemment. ROB_{loc} ne pénalise pas ce type d'altération, le centre de la boîte englobante ne bougeant pas. ROB_{cor} évalue le rapport hauteur/largeur et donne le plus mauvais résultat pour 90° tandis que ROB_{com}, qui pénalise la superficie occupée par la boîte englobante, donne le plus mauvais résultat pour 45° (voir Fig. 3.21).



(a) ROB_{cor} (b) ROB_{com}

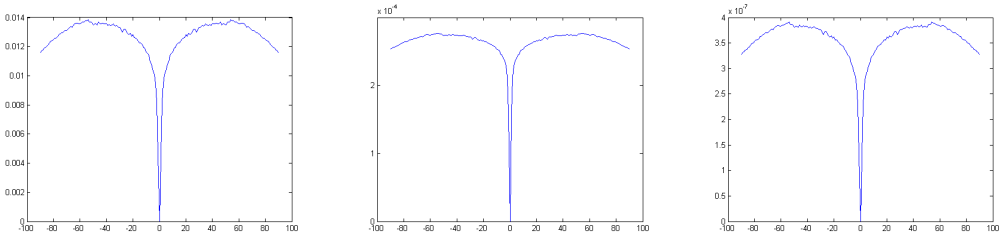
FIGURE 3.21 – Résultats des métriques ROB_{cor} et ROB_{com} pour une rotation sur la cinquième vérité terrain.

TABLE 3.5: Résultats pour l'altération de rotation

Métrique	Représentation	Symétrie d'axe Oz	Stricte monotonie	Régularité uniforme	Dépendance de taille	Dépendance de forme	Score
ROB_{loc}	Boîte						
ROB_{com}	Boîte	✓		✓	✓+/-		**
ROB_{cor}	Boîte	✓		✓		✓	***
$ErrLoc$	Contour	✓		✓	✓-		***
$ErrSous$	Contour	✓		✓	✓+/-		**
$ErrSur$	Contour	✓		✓	✓-		***
SNR	Contour	✓			✓+		**
RMS	Contour	✓		✓	✓-	✓	****
$Lq, 1$	Contour	✓		✓	✓-	✓	****
$Lq, 3$	Contour	✓			✓-	✓	***
KUL	Contour	✓		✓	✓-	✓	****
BAH	Contour	✓			✓+/-	✓	**
JEN	Contour	✓		✓	✓-	✓	****
$DMoy$	Contour	✓	✓	✓	✓-		****
$DMoC$	Contour	✓	✓	✓	✓-		****
FOM	Contour	✓	✓	✓	✓-	✓	*****
HAU	Contour	✓		✓	✓-		***
$BAD, 1$	Contour	✓	✓	✓	✓-	✓	*****
$BAD, 2$	Contour	✓	✓	✓	✓-		****
$BAD, 3$	Contour	✓	✓	✓	✓+/-	✓	****
$ODI_n, 1$	Contour	✓	✓	✓	✓-	✓	*****
$ODI_n, 2$	Contour	✓	✓	✓	✓-	✓	*****
$UDI_n, 1$	Contour	✓	✓	✓	✓-	✓	*****
$UDI_n, 2$	Contour	✓	✓	✓	✓-	✓	*****
PAS	Masque	✓	✓	✓	✓+/-		***
$HEN1$	Masque						
$HEN2$	Masque	✓	✓	✓	✓+/-		***
$YAS1$	Masque	✓	✓	✓	✓+/-		***
$YAS2$	Masque	✓	✓	✓	✓-		****
$YAS3$	Masque	✓	✓	✓	✓-		****
MAR_{gce}	Masque	✓	✓	✓	✓-	✓	*****
MAR_{lce}	Masque	✓	✓	✓	✓-	✓	*****
HAM	Masque	✓	✓	✓	✓-		****
$HAF1$	Masque	✓	✓	✓	✓-		****
$HAF2$	Masque	✓	✓	✓	✓-		****
VIN	Masque	✓	✓	✓	✓-		****
P_{px}	Masque	✓	✓	✓	✓+/-		***
R_{px}	Masque	✓	✓	✓	✓+/-		***

Les métriques basées contour $ErrLoc$, $ErrSous$, $ErrSur$, SNR, RMS, et les distances Lq, BAH, KUL, et JEN ne donnent pas des résultats satisfaisants pour la rotation non plus. En effet, on peut constater sur la figure 3.22 que ces métriques ne donnent pas le résultat le plus pénalisant pour 90°. De plus, on note que ces métriques sont très pénalisantes dès de très petites altérations.

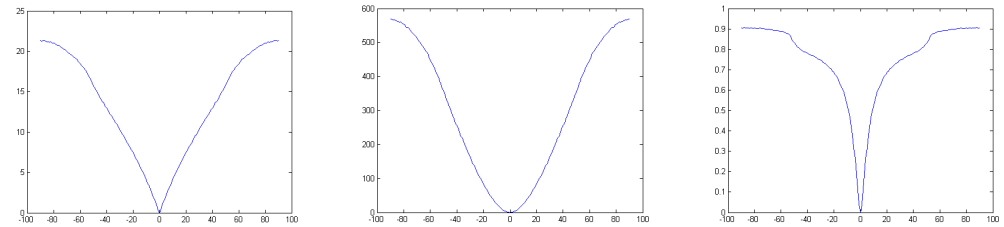
Les métriques $DMoy$, $DMoC$ et FOM présentées sur la figure 3.23 donnent ici



(a) $ErrLoc$ (b) RMS (c) JEN

FIGURE 3.22 – Résultats des métriques $ErrLoc$, RMS et JEN pour une rotation sur la cinquième vérité terrain.

aussi des résultats satisfaisants et pénalisent le plus lorsque la rotation est de 90°. Nous retrouvons également les différents comportements que nous avions observés avec les altérations précédentes. La métrique $DMoy$ pénalise quasiment linéairement la rotation, $DMoC$ est assez tolérante vis à vis des petites altérations tandis que FOM est elle assez sévère.

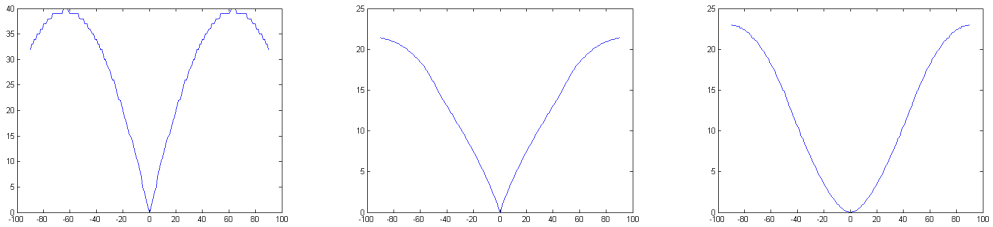


(a) $DMoy$ (b) $DMoC$ (c) FOM

FIGURE 3.23 – Résultats des métriques $DMoy$, $DMoC$ et FOM pour une rotation sur la cinquième vérité terrain.

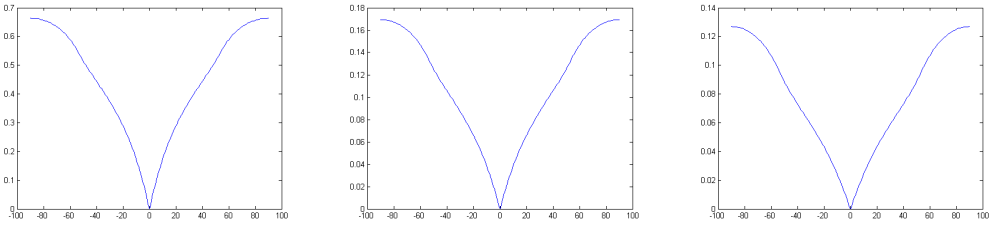
La distance de Hausdorff donne des résultats peu satisfaisants pour la rotation (voir Fig. 3.24). Les distances de Baddeley et les métriques d'Odet quant à elles donnent des résultats similaires et satisfaisants. Là encore, nous pouvons voir que le paramètre p de ces distances influe sur leurs comportements.

Enfin, l'ensemble des métriques basées sur des masques donnent des résultats corrects pour la rotation. En effet, toutes les métriques sont strictement monotones et continues avec le maximum de pénalisation à 90°. Nous pouvons voir à la figure 3.25 les résultats de trois métriques : PAS, $HAF, 2$ et VIN. Les autres métriques basées sur des masques ont des courbes similaires à celles-ci.



(a) HAU (b) $BAD, 1$ (c) $ODI_n, 2$

FIGURE 3.24 – Résultats des métriques HAU, $BAD, 1$ et $ODI_n, 2$ pour une rotation sur la cinquième vérité terrain.



(a) PAS (b) $HAF, 2$ (c) VIN

FIGURE 3.25 – Résultats des métriques PAS, $HAF, 2$ et VIN pour une rotation sur la cinquième vérité terrain.

Perspective

Nous pouvons voir à la table 3.6 les résultats obtenus pour l'altération de perspective. Nous remarquons tout d'abord que deux métriques sont insensibles à cette altération : ROB_{loc} et $HEN1$. De plus, la métrique BAH ne donne pas de bons résultats non plus. Enfin, nous pouvons voir que les métriques basées masque, à l'exception de $HEN1$, obtiennent le score maximal de 5, tandis que les métriques basées contour n'obtiennent pas mieux que 4.

Les métriques provenant du projet Robin n'obtiennent pas de bonnes performances ici. La métrique ROB_{loc} est même totalement insensible à ce genre d'altération, et ROB_{com} ne réagit pas correctement. On peut voir le défaut de ROB_{cor} qui n'est pas strictement monotone, notamment pour le cas $X = Y$ ou $X = -Y$.

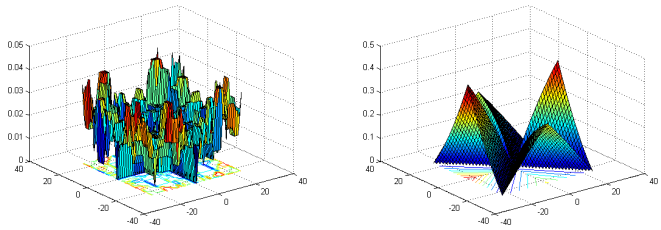
Les métriques basées contour n'obtiennent pas de très bons résultats non plus. En particulier, les métriques $ErrLoc$, $ErrSous$, $ErrSur$, SNR, RMS ainsi que les distances L_q, KUL et JEN comme on peut le voir à la figure 3.27. La métrique BAH réagit, pour sa part, très mal à cette altération en donnant des résultats similaires à

TABLE 3.6: Résultats pour l'altération de perspective

Métrique	Représentation	Symétrie d'axe Oz	Stricte monotonie	Régularité uniforme	Dépendance de taille	Dépendance de forme	Score
ROB_{loc}	Boîte						
ROB_{com}	Boîte				✓+/-	✓	*
ROB_{cor}	Boîte	✓		✓	✓+	✓	****
$ErrLoc$	Contour	✓		✓	✓-	✓	****
$ErrSous$	Contour	✓		✓	✓+/-	✓	***
$ErrSur$	Contour			✓	✓-	✓	***
SNR	Contour	✓			✓+	✓	***
RMS	Contour	✓			✓-	✓	***
$Lq, 1$	Contour	✓		✓	✓-	✓	****
$Lq, 3$	Contour	✓			✓-	✓	***
KUL	Contour	✓		✓	✓-	✓	****
BAH	Contour						
JEN	Contour	✓		✓	✓-	✓	****
$DMoy$	Contour			✓	✓+/-	✓	**
$DMoC$	Contour			✓	✓+/-	✓	**
FOM	Contour	✓		✓	✓+/-	✓	***
HAU	Contour	✓	✓	✓			***
$BAD, 1$	Contour			✓	✓+/-	✓	**
$BAD, 2$	Contour			✓	✓+	✓	***
$BAD, 3$	Contour			✓	✓+	✓	***
$ODI_n, 1$	Contour			✓	✓+/-	✓	**
$ODI_n, 2$	Contour			✓	✓+/-	✓	**
$UDI_n, 1$	Contour	✓		✓	✓+/-	✓	***
$UDI_n, 2$	Contour			✓	✓+/-	✓	**
PAS	Masque	✓	✓	✓	✓+	✓	*****
$HEN1$	Masque						
$HEN2$	Masque	✓	✓	✓	✓+	✓	*****
$YAS1$	Masque	✓	✓	✓	✓+	✓	*****
$YAS2$	Masque	✓	✓	✓	✓-	✓	*****
$YAS3$	Masque	✓	✓	✓	✓-	✓	*****
MAR_{gce}	Masque	✓	✓	✓	✓-	✓	*****
MAR_{lce}	Masque	✓	✓	✓	✓-	✓	*****
HAM	Masque	✓	✓	✓	✓-	✓	*****
$HAF1$	Masque	✓	✓	✓	✓+	✓	*****
$HAF2$	Masque	✓	✓	✓	✓-	✓	*****
VIN	Masque	✓	✓	✓	✓-	✓	*****
P_{px}	Masque	✓	✓	✓	✓+	✓	*****
R_{px}	Masque	✓	✓	✓	✓+	✓	*****

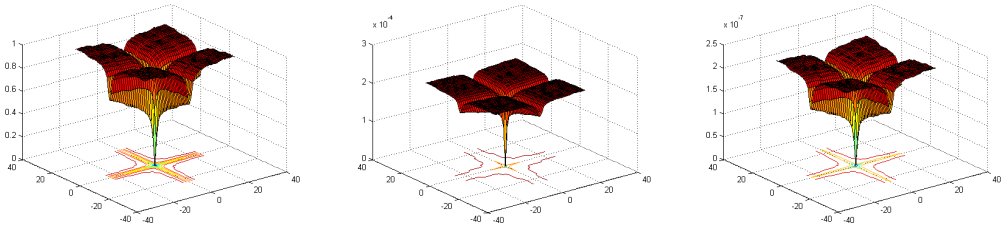
la métrique ROB_{com}.

Les autres métriques basées contour s'en sortent un peu mieux comme on peut le voir à la figure 3.28. Les métriques $DMoy$ et HAU ont un comportement similaire à la métrique $DMoC$ tandis que les mesures d'Odet et les distances de Baddeley se ressemblent.



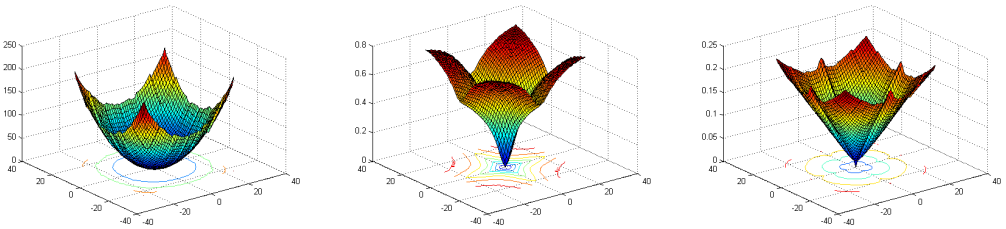
(a) ROB_{com} (b) ROB_{cor}

FIGURE 3.26 – Résultats des métriques ROB_{com} et ROB_{cor} pour une perspective sur la première vérité terrain.



(a) $ErrSous$ (b) RMS (c) JEN

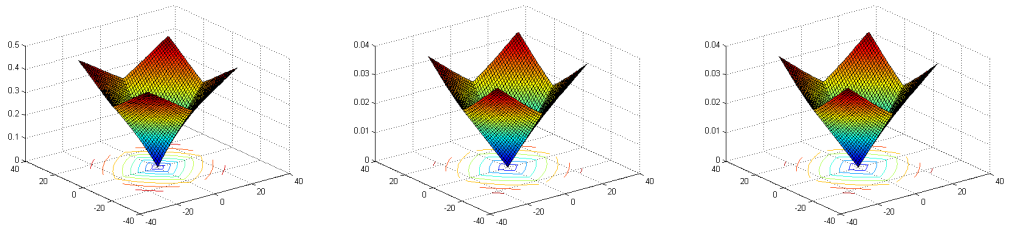
FIGURE 3.27 – Résultats des métriques $ErrSous$, RMS et JEN pour une mise perspective sur la première vérité terrain.



(a) $DMoC$ (b) FOM (c) $BAD, 3$

FIGURE 3.28 – Résultats des métriques $DMoC$, FOM et $BAD, 3$ pour une mise perspective sur la première vérité terrain.

Mis à part la métrique $HEN, 1$, l'ensemble des métriques basées masque ont un bon comportement face à l'altération de mise en perspective, comme on peut le voir à la figure 3.29.



(a) PAS (b) MAR_{gce} (c) VIN

FIGURE 3.29 – Résultats des métriques PAS, MAR_{gce} et VIN pour une mise perspective sur la première vérité terrain.

3.2.3 Discussions

Nous avons rassemblé l'ensemble de résultats obtenus dans le tableau 3.7. De plus, nous avons fait la somme de scores obtenus pour chaque altération afin d'obtenir un score final pour chaque métrique, dont la note maximale est de 22.

Tout d'abord, nous pouvons voir que les trois métriques utilisant les boîtes englobantes ne sont pas en mesure d'évaluer correctement un résultat de localisation. Cependant, ces résultats sont à nuancer car ces trois métriques ont été créées dans le but d'être utilisées ensemble avant d'obtenir un résultat final. Cependant, vérifier si les propriétés sont remplies par les trois métriques utilisées conjointement, comme dans la compétition Robin, n'aurait pas de sens. En effet, on a vu qu'un seuillage est effectué sur chacune des trois métriques. La métrique finale, donnant un résultat binaire 0 ou 1, ne remplirait alors aucune propriété.

Ensuite, nous avons constaté que les métriques utilisant une représentation en contour donnent de moins bons résultats que les métriques se basant sur une représentation par des masques. En effet, si l'on excepte la métrique $HEN1$, toutes ces métriques obtiennent une note supérieure à 16. La majorité des métriques se basant sur une représentation en contour n'obtiennent pas, quant à elles, une note supérieure à 16. Seules les métriques $DMoy$, $DMoC$, FOM et BAD dépassent cette note de 16. Cela nous amène à recommander l'utilisation des métriques se basant sur des représentations par des masques, et en particulier les métriques définies par Martin *et coll.* dans [47].

TABLE 3.7: Synthèse des résultats obtenus

Métrique	Représentation	Distance	Translation	Changement d'échelle	Rotation	Perspective	Score final
ROB_{loc}	Boîte		*****				5
ROB_{com}	Boîte	*		***	**	*	7
ROB_{cor}	Boîte	*		***	***	****	11
$ErrLoc$	Contour	***	***	**	***	****	15
$ErrSous$	Contour	**	**	*	**	***	10
$ErrSur$	Contour	**	***	**	***	***	13
SNR	Contour		***	**	**	***	10
RMS	Contour	***	***	**	****	***	15
$Lq, 1$	Contour	***	***	**	****	****	16
$Lq, 3$	Contour	***	***	**	***	***	14
KUL	Contour	**	***	**	****	****	15
BAH	Contour	*	***	****	**	—	10
JEN	Contour	**	***	**	****	****	15
$DMoy$	Contour	**	*****	***	****	**	16
$DMoC$	Contour	**	*****	***	****	**	16
FOM	Contour	**	****	***	*****	***	17
HAU	Contour	***	***	**	***	***	14
$BAD, 1$	Contour	***	****	**	*****	**	16
$BAD, 2$	Contour	***	***	***	****	***	16
$BAD, 3$	Contour	***	***	***	****	***	16
$ODI_n, 1$	Contour	**	**	*	*****	**	12
$ODI_n, 2$	Contour	**	***	*	*****	**	13
$UDI_n, 1$	Contour	**	**	*	*****	***	13
$UDI_n, 2$	Contour	**	***	*	*****	**	13
PAS	Masque	***	*****	****	***	*****	**20**
$HEN1$	Masque			***			3
$HEN2$	Masque	**	*****	****	***	*****	19
$YAS1$	Masque		*****	***	***	*****	16
$YAS2$	Masque		*****	***	****	*****	17
$YAS3$	Masque		*****	***	****	*****	17
$\mathbf{MAR_{gce}}$	Masque	***	*****	****	*****	*****	**22**
$\mathbf{MAR_{lce}}$	Masque	***	*****	****	*****	*****	**22**
HAM	Masque	***	****	***	****	*****	19
$HAF1$	Masque	***	****	*	****	*****	17
$HAF2$	Masque	**	****	***	****	*****	18
VIN	Masque	***	*****	****	****	*****	**21**
P_{px}	Masque		*****	***	***	*****	16
R_{px}	Masque		*****	***	***	*****	16

3.3 Évaluation du modèle pour la reconnaissance

Tout comme pour la localisation, nous faisons ici ressortir les métriques de l'état de l'art permettant de pénaliser les erreurs de reconnaissance d'objets. Cependant, contrairement à la localisation, on ne peut pas calculer une distance sur les identifiants numériques ou les chaînes de caractères représentant l'objet reconnu par l'algorithme. De plus, l'erreur commise en affectant la classe « chien » à un objet de la classe

« chat » peut être calculée préalablement à l'évaluation des résultats de l'algorithme constituant un modèle de pénalisation d'erreurs.

Nous nous sommes donc intéressés aux méthodes permettant de calculer une distance sur les modèles représentant les objets. Plus exactement, notre but est de calculer une mesure de similarité entre les classes d'objets telle que la similarité entre deux objets d'une même catégorie soit plus importante que celle entre deux objets de catégories différentes. Pour cela, nous avons d'abord comparé les différents modèles permettant de représenter les objets. Nous avons ensuite comparé différents descripteurs de points d'intérêt et quantifié leur capacité à catégoriser des objets.

Dans cette première partie, nous nous sommes intéressés à différentes méthodes de calcul de distances entre des objets en se basant sur deux représentations différentes de ces objets : les ensembles de points et les graphes. Les ensembles de points sont le mode de représentation le plus largement utilisé dans la littérature [24, 70, 10, 71]. Les graphes ont, quant à eux, l'avantage de reprendre les informations contenues dans les ensembles de points et d'y ajouter des données concernant la proximité des points entre eux, c'est-à-dire des informations spatiales.

3.3.1 Protocole

Calcul de distance entre modèles

Nous nous sommes intéressés principalement à deux modèles : les ensembles de points et les graphes.Les données extraites afin de construire les ensembles de points ainsi que les graphes ont été obtenues avec le détecteur de points d'intérêt et le descripteur SIFT. Le descripteur SIFT est utilisé comme descripteur de référence et ne présume pas du choix que nous allons faire par la suite pour le modèle d'un objet. Cette étude comparative a été menée sur des objets particuliers à savoir des visages d'individus.

Ensembles de points

Afin de calculer une distance entre deux ensembles de points, nous effectuons le calcul d'associations proposé dans [72]. Dans cet article, une association est définie en tant que double mise en correspondance entre deux points d'intérêt. Cette mise en correspondance a été proposée pour la mise en correspondance des descripteurs SIFT dans [25]. Pour le point d'intérêt x de l'image k, nous recherchons le point d'intérêt y le plus proche parmi l'ensemble $Y(I_l)$ de tous les points d'intérêts de l'image l. Nous

regardons également si le second point d'intérêt y' le plus proche est suffisamment loin de x au moyen d'une valeur seuil C :

$$d(x, y) = \min_{z \in Y(I_l)} d(x, z) \tag{3.1}$$

et

$$d(x, y) \leq C * d(x, y') \tag{3.2}$$

la distance $d(\cdot, \cdot)$ est une distance euclidienne calculée entre les deux descripteurs normalisés correspondant aux points d'intérêt. Si ces deux conditions ne sont pas remplies, alors le point x n'est pas mis en correspondance avec le point y.

Nous définissons alors une association entre les points x et y comme une double mise en correspondance ; c'est-à-dire lorsque le point x est mis en correspondance avec le point y et le point y est mise en correspondance avec le point x. La similarité entre les deux ensembles de points est tout simplement le nombre de points d'intérêt mis en correspondance.

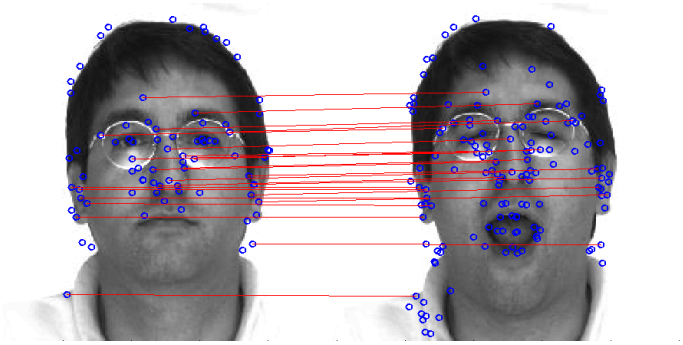


FIGURE 3.30 – Exemple de double associations : les lignes relient les points associés dans chacune des images

Graphes

Nous construisons un graphe à partir de l'ensemble de points obtenus précédemment. Le calcul du graphe à partir de ces points peut se faire de différentes façons [73] :

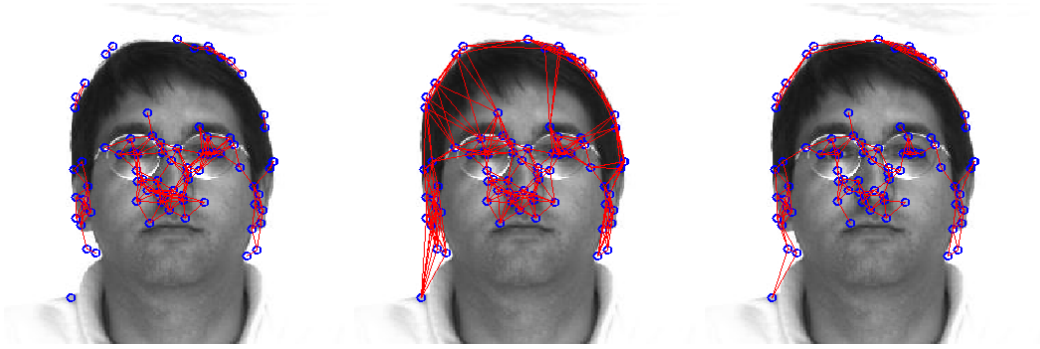
le graphe du ϵ-voisinage : dans ce cas, nous connectons ensemble les points qui sont séparés au maximum par une distance ϵ,

le graphe des k-plus proches voisins : dans ce graphe, nous connectons le nœud v_i au nœud v_j si celui-ci appartient aux k-plus proches voisins de v_i. Cela mène à un graphe orienté et deux possibilités peuvent le rendre non orienté :

soit l'on considère l'arête sans son orientation, soit l'on considère qu'il faut que v_i appartienne également aux k-plus proches voisins de v_j. On parlera respectivement de graphe symétrique et de graphe mutuel,

le graphe complet : ce graphe connecte tous les nœuds ensemble. De par sa complexité, nous n'utiliserons pas ce graphe.

La figure 3.31 présente un exemple de chacun des graphes que nous utilisons par la suite. Nous pouvons remarquer que le graphe mutuel est un sous-graphe du graphe symétrique.



(a) ϵ-voisinage (b) Symétrique (c) Mutuel

FIGURE 3.31 – Exemples de graphes, les nœuds sont représentés par des ronds bleus et les arêtes par des lignes rouges

Pour les graphes, nous nous sommes intéressés à trois calculs différents de distance. Le premier d'entre eux est un calcul de distance euclidienne effectué directement sur la matrice d'adjacence. Cependant, afin de comparer deux matrices d'adjacence, celles-ci ont besoin d'avoir la même taille, il doit donc y avoir le même nombre de nœuds dans les graphes. Nous avons alors calculé la double mise en correspondance présentée ci-dessus, puis calculé le graphe sur les points sélectionnés.

La seconde méthode que nous avons utilisée est la distance fournie par l'algorithme de Munkres [53], dont la matrice des coûts a été calculée à partir de la distance euclidienne calculée sur les vecteurs descripteurs SIFT. La troisième méthode utilisée est l'algorithme de Bunke [54], qui fournit également une bonne approximation de la distance d'édition.

Base de données

Afin d'évaluer les performances de ces différentes représentations et mesures de similarité, nous nous sommes placés dans le contexte particulier de la biométrie. La biométrie consiste à identifier ou authentifier des individus à partir de la reconnaissance de caractéristiques qui leur sont propres. La biométrie est donc un type particulier d'algorithme de reconnaissance, et donc d'interprétation. De plus, la biométrie est un thème de recherche important dans le laboratoire GREYC.

Nous avons utilisé la base de visages AR [1] [74]. Cette base comprend les visages de 126 personnes, 70 hommes et 56 femmes, et dispose de 26 prises de vue par individus. Ces 26 prises de vue ont été réalisées au cours de deux sessions, la seconde ayant lieu deux semaines après la première. Lors de chaque session, 13 photos ont été prises sous différentes conditions d'expression (neutre, souriant, en colère, criant), d'illumination (lumière venant de la droite, de la gauche) et d'occultation (portant des lunettes de soleil, une écharpe). Des exemples de ces images sont donnés à la figure 3.32.

Cependant, les données récupérées ne comportaient pas les 26 images par personne. Nous avons donc éliminé les individus ne disposant que d'une seule session. Nous avons alors obtenu une base de visages comportant 120 personnes, 65 hommes et 55 femmes, et de 26 photos par personnes, soit *3 120* photos. Ces photos sont des images couleurs au format raw de $768 * 576$ pixels. Les images ont alors été recadrées en $576 * 576$ pixels puis réduites en $256 * 256$ pixels et converties en niveau de gris.

Mesure de performance

Dans nos expérimentations, nous avons utilisé la première image de chaque individu comme référence. Nous avons ensuite calculé la distance aux 25 autres images de cet individu, ce qui constitue l'ensemble de données intraclasses (Intra) tandis que la distance calculée par rapport aux 26 images des 119 individus restants constitue l'ensemble des données interclasses (Inter).

À partir de ces deux ensembles, nous avons pu calculer les courbes DET ainsi que le taux d'égale erreur (EER) et l'aire sous la courbe (AUC) que nous avons présentés dans le chapitre précédent (voir section 2.6.1). Plus ces mesures sont faibles, plus la méthode est performante.

1. http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html

(a) Neutre

(b) Expressions

(c) Illuminations

(d) Occultations

FIGURE 3.32 – Exemple d'une session d'acquisition de visages provenant de la base AR [74]

3.3.2 Résultats

Paramétrage de la double association

Nous avons commencé par regarder les performances de la double association en fonction du paramètre C, valeur seuil de la double association. Le taux d'égale erreur et l'aire sous la courbe sont présentés dans le tableau 3.8.Nous pouvons voir que les meilleures performances sont obtenues pour $C = 0,7$. En effet, c'est pour cette valeur que le taux d'égales erreur ainsi que l'aire sous la courbe sont minimisés.

TABLE 3.8: Performance de la double association

C	Taux d'égale erreur	Aire sous la courbe
0,6	16,86%	10,02%
0,65	16,72%	9,33%
0,7	**16,37**%	**9,28**%
0,75	16,49%	9,45%
0,8	17,02%	9,60%

Modèle de graphe utilisé

Nous avons ensuite regardé quel était le modèle de graphe le plus adapté à nos données. Pour cela, nous avons comparé les performances obtenues avec la méthode utilisant la matrice d'adjacence. Le paramètre C de la double association nécessaire pour cette méthode a été fixé à 0,7. Le tableau 3.9 présente le taux d'égale erreur pour différents modèles de graphe.

TABLE 3.9: Performance pour différents modèle de graphes

Type de graphes	Paramètre	Taux d'égale erreur	Aire sous la courbe
ϵ-voisinage	$\epsilon = 10$	21,82%	14,51%
ϵ-voisinage	$\epsilon = 20$	17,11%	11,40%
ϵ-voisinage	$\epsilon = 30$	**16,17**%	**10,68**%
Symétrique	$k = 3$	16,82%	11,19%
Symétrique	$k = 5$	18,62%	11,87%
Symétrique	$k = 7$	19,28%	12,37%
Mutuel	$k = 3$	16,82%	11,09%
Mutuel	$k = 5$	18,62%	11,86%
Mutuel	$k = 7$	19,28%	12,37%

Nous pouvons constater que les meilleures performances sont obtenues avec les graphes de types ϵ-voisinage. Le paramètre $\epsilon = 30$ permet de baisser le taux d'erreur à 16,17%, ce qui est légèrement meilleur que la double association seul. Les autres méthodes, type de graphe ou paramétrage, détériorent les performances par rapport à la double association seul qui était de 16,37%. On constate également que les graphes symétriques obtiennent des performances similaires aux graphes mutuels.

Performances des méthodes de calcul de la distance d'édition

Nous avons ensuite calculé les performances avec la méthode de l'algorithme Munkres ainsi que celle de l'algorithme de Bunke qui fournissent une approximation de la distance d'édition. Nous avons utilisé pour cela des graphes de types ϵ-voisinage, avec $\epsilon = 30$, compte tenu des résultats précédents. Les performances étaient alors respectivement de $39,32\%$ et $50,81\%$. Ces résultats n'étant pas satisfaisants, nous avons calculé les performances de ces algorithmes sur un graphe réduit grâce à la

double association en prenant $C = 0,7$. Nous avons pour cela fait évoluer le paramètre α qui fait varier la prise en compte du voisinage dans l'algorithme de Bunke. Le tableau 3.10 présente le taux d'égale erreur ainsi que l'aire sous la courbe.

TABLE 3.10: Performance des méthodes de calcul de la distance d'édition

Métrique	Paramètre	Taux d'égale erreur	Aire sous la courbe
Munkres		16,88%	**9,71**%
Bunke	$\alpha = 0,2$	**16,61**%	9,94%
Bunke	$\alpha = 0,4$	16,79%	10,11%
Bunke	$\alpha = 0,6$	16,82%	10,22%
Bunke	$\alpha = 0,8$	16,91%	10,32%

Nous pouvons voir que l'algorithme de Bunke n'améliore pas les performances par rapport à l'algorithme de Munkres dans notre cas. En effet, l'aire sous la courbe est minimale pour l'algorithme de Munkres, tandis que le meilleur taux d'égale erreur est obtenu pour une faible valeur de α. Plus la valeur de α est faible, moins les voisins sont pris en compte. De fait, si l'on met α à 0, on retrouve l'algorithme de Munkres.

L'utilisation de ces algorithmes ne permet pas d'améliorer les performances obtenus par la seule double association. De plus, l'utilisation de l'algorithme de Munkres sur le graphe réduit revient à calculer la somme des coûts des points associés par la double association.

Temps de calcul

Nous avons également mesuré le temps de calcul des différentes méthodes. Nous présentons les résultats dans le tableau 3.11.

TABLE 3.11: Temps de calcul des différentes méthodes

Métrique	Temps de calcul entre deux images (ms)	Temps de calcul entre deux classes (s)
Double association	32	0,84
Matrice	32	0,85
Munkres	32	0,85
Bunke	39	1,02

Nous pouvons constater que, mis à part l'algorithme de Bunke, les temps de calcul des méthodes sont proches. En effet, toutes les méthodes nécessitent au préalable au moins une double association. Seul l'algorithme de Bunke fait des calculs supplémentaires en s'intéressant au voisinage des points sélectionnés, ce qui explique qu'il est légèrement plus long à calculer.

3.3.3 Discussions

Nous avons présenté les résultats de toutes les méthodes testées avec le meilleur paramétrage à la figure 3.33.

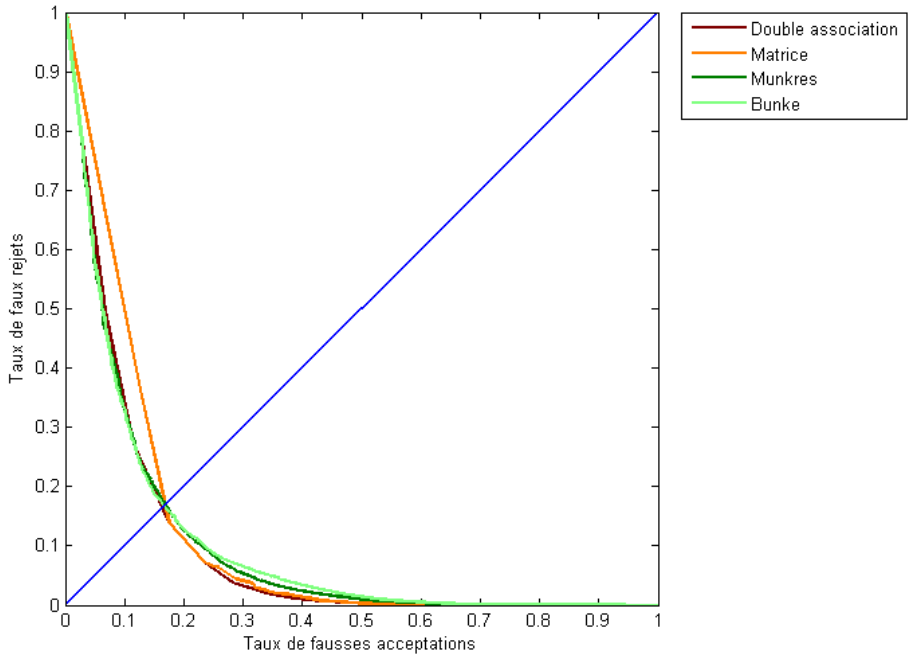


FIGURE 3.33 – Récapitulatif des performances

Nous présentons également le taux d'égale erreur et l'aire sous la courbe à la table 3.12.

TABLE 3.12: Récapitulatif des performances

Métrique	Taux d'égale erreur	Aire sous la courbe
Double association	16,37%	**9,28**%
Matrice	**16,17**%	10,68%
Munkres	16,88%	9,71%
Bunke	16,61%	9,94%

Nous pouvons voir que toutes les méthodes obtiennent des performances similaires. De plus, on peut constater que l'algorithme de Bunke n'améliore pas les performances de l'algorithme de Munkres, ni même celles de la double association. Suite à ces tests, nous constatons que l'utilisation des graphes n'apportent pas d'amélioration des performances lors de la reconnaissance. Parmi les différents types disponibles, le graphe ϵ-voisinage est à privilégier si l'on souhaite utiliser des graphes. Cependant,

les résultats des méthodes basées graphes sont conditionnés par la double association des points au préalable. En effet, si l'on ne fait pas cette double association, les performances tombent à plus de 40% de taux d'égale erreur. L'utilisation des graphes entraîne alors une double association préalable, ce qui empêche alors tout gain de temps de calcul, comme nous avons pu le voir.

En conclusion, nous utiliserons pour mesurer la similarité d'objets, la double association avec $C = 0,7$. Nous avons vu que les points d'intérêt ont été décrit au moyen du descripteur SIFT. Si ce descripteur est largement reconnu par la communauté, nous le comparons néanmoins dans la section suivante avec d'autres descripteurs.

3.4 Évaluation des descripteurs pour la reconnaissance

Nous visons à mesurer une similarité entre deux classes ou bien entre deux catégories d'objets. Nous voulons pondérer une erreur de reconnaissance en considérant la similarité des classes. Nous supposons ici que plus le nombre de points d'intérêts correspondants entre deux images d'objets de classes différentes est important, plus ces classes sont similaires. La figure 3.34 présente un exemple de points d'intérêts correspondants entre deux images. Nous pouvons voir que ces images, bien que provenant des classes « calculatrice » et « clavier d'ordinateur », présentent certaines similitudes notamment par la présence de boutons sur les deux objets. Ces deux classes appartiennent à une même catégorie « électronique ».

FIGURE 3.34 – Exemple de points correspondants entre deux images de classes différentes, appartenant à une même catégorie

Le calcul de cette mesure de similarité entre deux objets se fait en deux étapes.

Tout d'abord, nous détectons les points qui s'associent entre deux images différentes comme dans la section précédente, ce qui nous donne un score de similarité entre deux images. Ensuite, nous calculons une mesure de similarité entre deux classes à partir du score de similarité sur l'ensemble des images de chacune de ces classes.

3.4.1 Protocole

Mesures de similarité

Nous définissons la mesure de similarité $s(k,l)$ entre deux images k et l en tant que nombre de points d'intérêt associés par la double association présentée précédemment. Plus le nombre d'associations est important et plus les images contiennent des objets similaires. À partir de cela, nous avons calculé deux mesures de similarité entre objets à partir de plusieurs images représentant ces objets. La première prend simplement en compte le nombre d'associations. La seconde calcule le nombre d'associations normalisé par le nombre de points d'intérêt dans les images. Pour obtenir la première mesure de similarité $S_1(i,j)$ entre deux objets i et j, nous calculons la similarité moyenne entre les images de ces objets :

$$S_1(i,j) = \begin{cases} \frac{1}{Card(i)*Card(j)} \sum_{k\in i} \sum_{l\in j} s(k,l) & \text{si } i \neq j \\ \frac{2}{(Card(i)-1)*Card(i)} \sum_{k\in i} \sum_{l>k\in i} s(k,l) & \text{sinon} \end{cases} \tag{3.3}$$

avec $Card(i)$ le nombre d'images pour l'objet i. Nous pouvons remarquer que plus les objets seront similaires, et plus la similarité sera importante. De plus, nous avons $S_1(i,j) = S_1(j,i)$.

Pour la seconde mesure de similarité $S_2(i,j)$, nous avons décidé de normaliser par le nombre de points d'intérêt dans les images dont on calcule la similarité. En effet, un grand nombre de points d'intérêt mis en correspondance est plus pertinent dans le cas où le nombre de points d'intérêt présents dans les images est plus faible. Nous avons alors défini la mesure suivante :

$$S_2(i,j) = \begin{cases} \frac{1}{Card(i)*Card(j)} \sum_{k\in i} \sum_{l\in j} \frac{s(k,l)}{\min(N_k,N_l)} & \text{si } i \neq j \\ \frac{2}{(Card(i)-1)*Card(i)} \sum_{k\in i} \sum_{l>k\in i} \frac{s(k,l)}{\min(N_k,N_l)} & \text{sinon} \end{cases} \tag{3.4}$$

avec N_k le nombre de points d'intérêt dans l'image k.

Afin de valider nos résultats, nous avons calculé les courbes DET ainsi que le taux d'égale erreur comme précédemment. Pour cela, nous avons calculé les

ensembles INTER et INTRA nécessaires au tracage des courbes DET. Dans le cas de la reconnaissance d'objets, les ensembles sont définis ainsi :

$$\begin{aligned} \text{INTRA}_{objets} &= \{S(i,i)|\forall i\} \\ \text{INTER}_{objets} &= \{S(i,j)|\forall i,j,i \neq j\} \end{aligned} \tag{3.5}$$

Concernant la performance en catégorisation, nous avons construit les ensembles suivants :

$$\begin{aligned} \text{INTRA}_{categories} &= \{S(i,j)|\forall u,\forall i,j \in u^2, i \neq j\} \\ \text{INTER}_{categories} &= \{S(i,j)|\forall u,v,u \neq v,\forall i \in u,\forall j \in v\} \end{aligned} \tag{3.6}$$

Enfin, cela nous permet de définir une matrice de similarité MS. Nous définissons la matrice de similarité telle que $MS_{i,j}$ soit égale à la similarité entre les objets i et j, c'est-à-dire $MS_{i,j} = S(i,j)$. Cette matrice ressemble à une matrice de confusion mais en diffère par plusieurs points. Tout d'abord, cette matrice est symétrique car $S(i,j) = S(j,i)$. Ensuite, cette matrice est construite comme outil d'évaluation, indépendamment des résultats des algorithmes que nous souhaitons évaluer. En effet, une fois construite, cette matrice va permettre de prendre en compte l'importance de l'erreur commise en reconnaissant l'objet i à la place de l'objet j.

	Chat	Chien	Voiture	Bus	Moto
Chat	0	0,1	1	1	1
Chien	0,1	0	1	1	1
Voiture	1	1	0	0,2	0,2
Bus	1	1	0,2	0	0,3
Moto	1	1	0,2	0,3	0

TABLE 3.13: Exemple de matrice de distance entre classes construite manuellement

Points d'intérêts

Nous avons calculé les points d'intérêt préalablement au calcul de descripteurs. Cela permet de s'assurer que tous les descripteurs sont calculés sur les mêmes points d'intérêt. Pour cela, nous avons utilisé le détecteur Harris Affine présenté dans [71]. Nous avons décidé d'utiliser ce descripteur car il permet de détecter plus de points d'intérêt que la différence de Gaussienne utilisée dans SIFT. De fait, nous avons fait varier le seuil du détecteur afin d'obtenir N points d'intérêt, avec $0.8 * \frac{Largeur(I)*Hauteur(I)}{2000} < N < 1.2 * \frac{Largeur(I)*Hauteur(I)}{2000}$.

Descripteurs

Nous avons décidé d'utiliser les descripteurs suivants :
- les filtres complexes [26],
- les filtres « Steerable » [27],
- les invariants différentiels [28],
- SIFT [25],
- ACP-SIFT [30],
- le contexte de forme [31],
- l'histogramme des orientations et des gradients [4].

Il faut noter que pour le descripteur ACP-SIFT, le calcul des points d'intérêt est normalement identique à celui de SIFT, tandis que le calcul du descripteur est une ACP sur l'imagette du voisinage. Cependant, nous avons calculé les points d'intérêt indépendamment du vecteur descripteur. De fait, le descripteur ACP-SIFT correspond alors à une simple ACP. Nous avons également ajouté à ces descripteurs l'imagette du voisinage redimensionnée en 9*9, ce qui nous amène à un vecteur descripteur de 81 éléments.

Base d'images

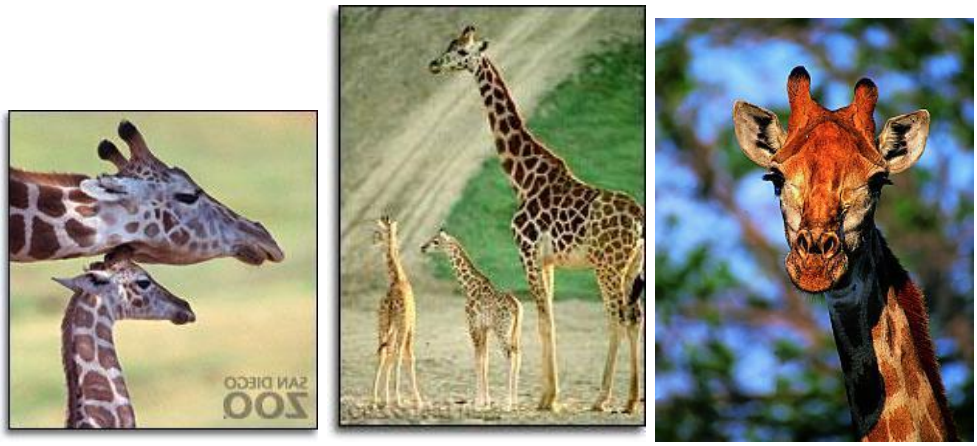
Pour cette étude comparative, nous utilisons les images de la base Caltech256 [2] [1]. Cette base de données présente de nombreux avantages. Le premier d'entre eux est le nombre important de classes d'objets dans la base ainsi que leurs diversités : 256 classes d'objets telles que « sac à dos », « machine à pain » ou « girafe ». Cependant, la diversité de ces classes ainsi que la variété des images de chaque classe rend les classes de cette base de données difficile à séparer. Des exemples d'images de cette base de données sont présentés à la figure 3.35. Le nombre d'images pour chaque classe est variable et est compris entre 80 et 827.

Le second avantage de cette base d'images est que les classes sont ordonnées selon une taxonomie, voir figure 3.36. Cette taxonomie crée des catégories de classes d'objets, chaque classe d'une catégorie comportant des objets sémantiques proches.

Enfin, un troisième avantage à cette base de données est qu'elle présente la difficultés de reconnaissance des classes. Ainsi, les classes « avion », « montre » et « moto » sont plus faciles à reconnaître que les classes « chien »,« fusil » et « skateboard ». Le

2. http://www.vision.caltech.edu/Image_Datasets/Caltech256/

(a) Classe « sac à dos »



(b) Classe « girafe »

FIGURE 3.35 – Exemple d'images tirées de la base CalTech 256

taux de bonne reconnaissance sur la base entière est en moyenne d'environ 35% dans la littérature, comme on peut le voir à la figure 3.37, ce qui montre bien la difficulté de ce benchmark.

Nous avons utilisé une partie de la base de données. Nous avons choisi 64 classes d'objets, soit un quart de la base totale. Ces 64 classes d'objets sont équitablement réparties parmi la difficulté de reconnaissance. Pour 64 classes, le taux de bonne reconnaissance attendu dans [1] est d'environ 45% à 55%. La taxonomie des classes sélectionnées est présentée à la figure 3.38. Nous avons pris les 20 premières images de chaque classe d'objets, ce qui nous amène à *1 280* images et *818 560* mesures de similarité entre images pour chaque descripteur et paramètre testés.

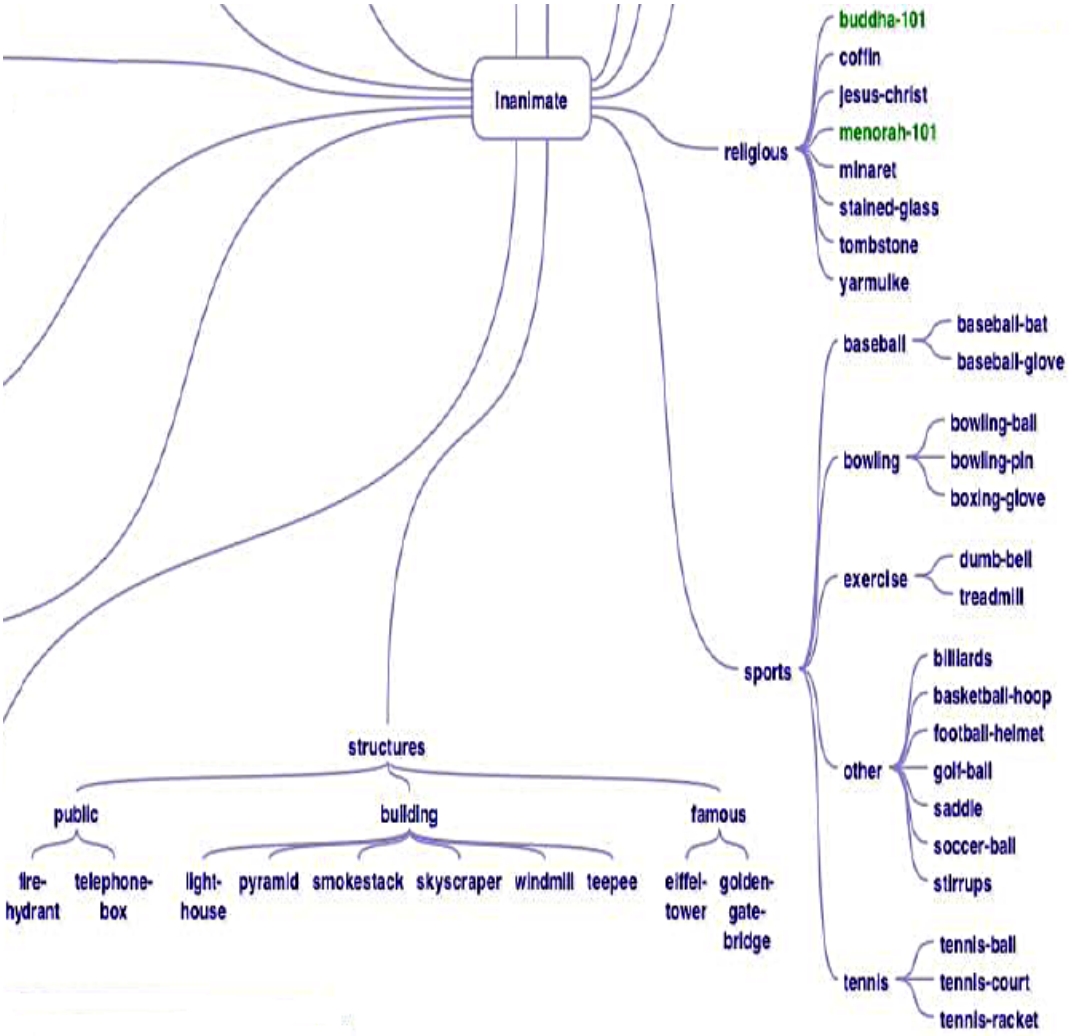


FIGURE 3.36 – Extrait de la base d'images Caltech256 [1] : les feuilles de l'arbre représentent les classes d'objets tandis que les nœuds représentent des catégories.

3.4.2 Résultats

Reconnaissance

Nous présentons tout d'abord les résultats pour la reconnaissance des classes. La figure 3.39 présente les courbes DET pour les 8 descripteurs et les 2 scores testés.

Le taux d'égale erreur est présenté dans le tableau 3.14. Nous pouvons voir tout d'abord que les résultats de reconnaissance ne sont pas très bons (entre 35% et 45% d'erreur). Ils sont néanmoins meilleurs que ceux obtenus dans [1]. Nous remarquons ensuite que la normalisation n'apporte pas une amélioration des performances dans ce cas, à l'exception de deux descripteurs : les filtres complexes et les invariants

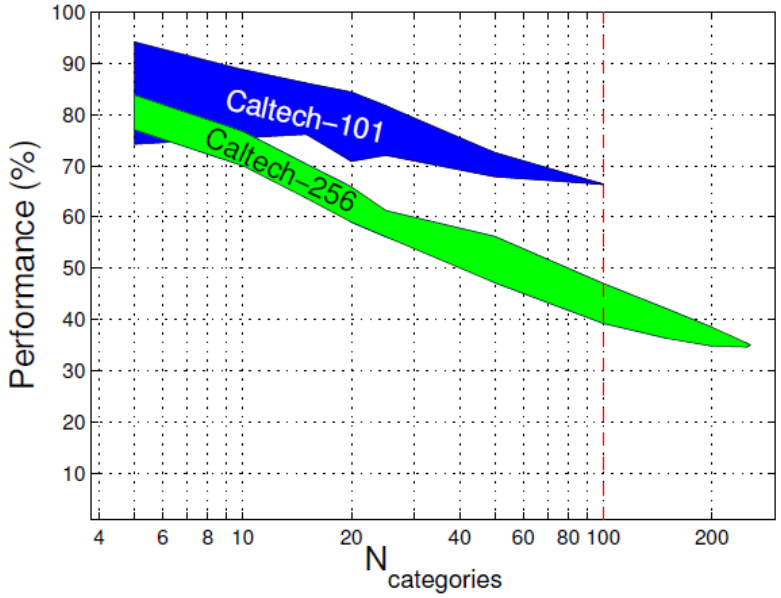


FIGURE 3.37 – Performance en fonction du nombre de classes utilisées [1]

différentiels. Nous pouvons remarquer que de bonnes performances sont obtenues par les filtres Steerable, le descripteur SIFT, l'ACP ainsi que par l'utilisation d'une imagette.

TABLE 3.14: Résultats pour la reconnaissance de classes

Descripteurs	Nombre d'associations	Normalisation
Filtres complexes	41.76 %	40.12 %
Filtres Steerable	34.51 %	**38.47 %**
Invariants différentiels	41.98 %	41.65 %
SIFT	35 %	42.59 %
ACP	**33.57 %**	41.06 %
Contexte de forme	36.73 %	44.84 %
Histogramme	37.73 %	42.47 %
Imagette	35.16 %	42.63 %

Catégorisation

Nous avons ensuite fait les mêmes calculs pour la catégorisation. Les courbes DET sont présentées à la figure 3.40. Nous avons également calculé les taux d'égale erreur dans le cas de la catégorisation, présentés dans le tableau 3.15.

Nous remarquons tout d'abord une légère perte de performance par rapport à la reconnaissance, d'environ 10%. Nous remarquons ensuite que la normalisation n'apporte, en général, pas d'amélioration par rapport aux nombres d'associations,

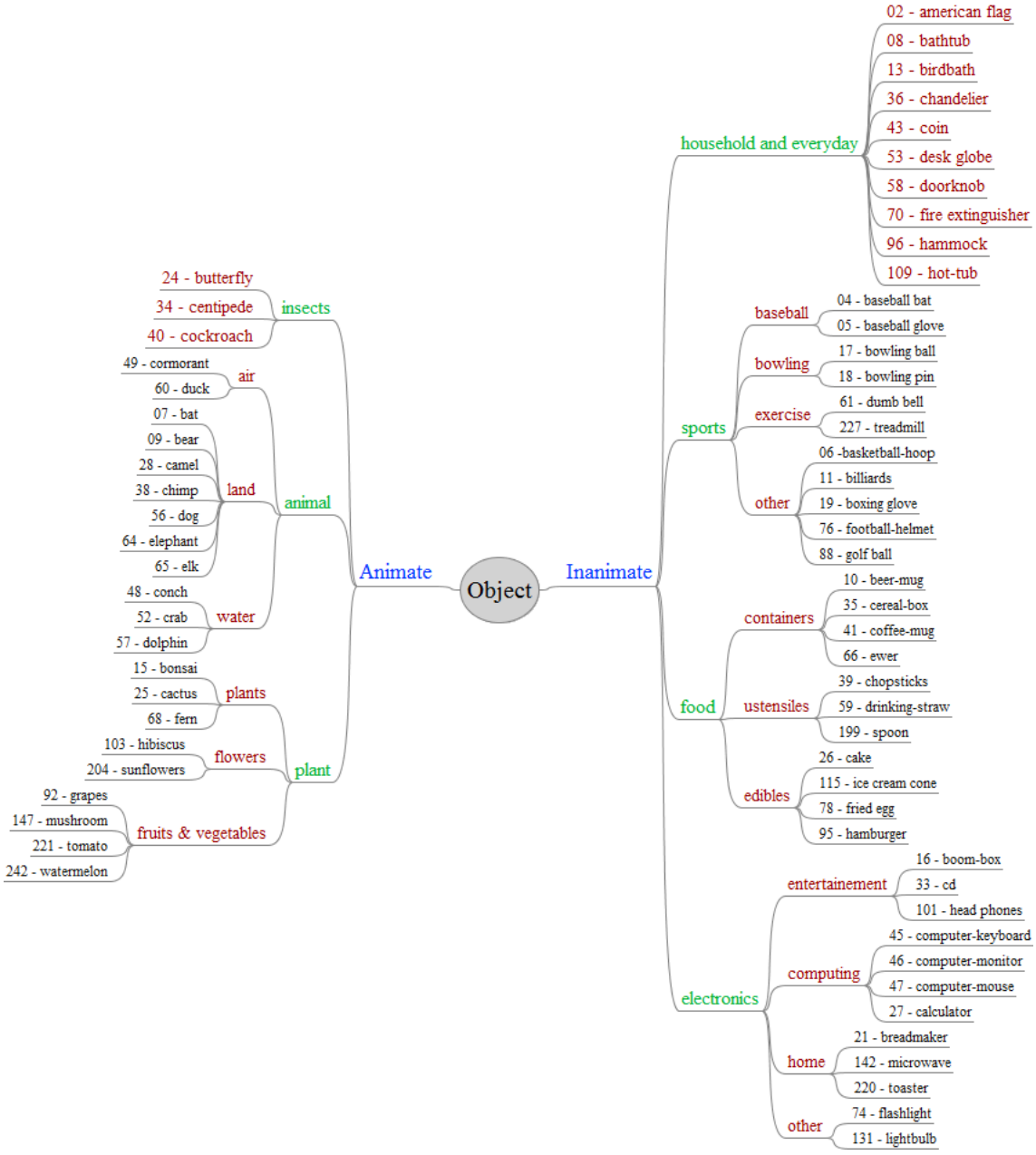
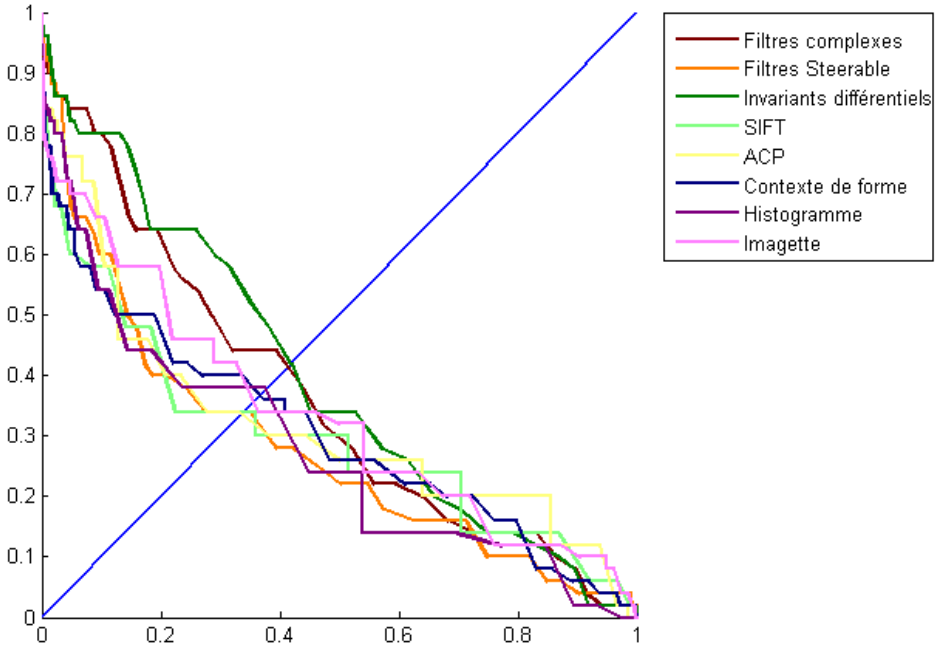


FIGURE 3.38 – Taxonomie des classes sélectionnées

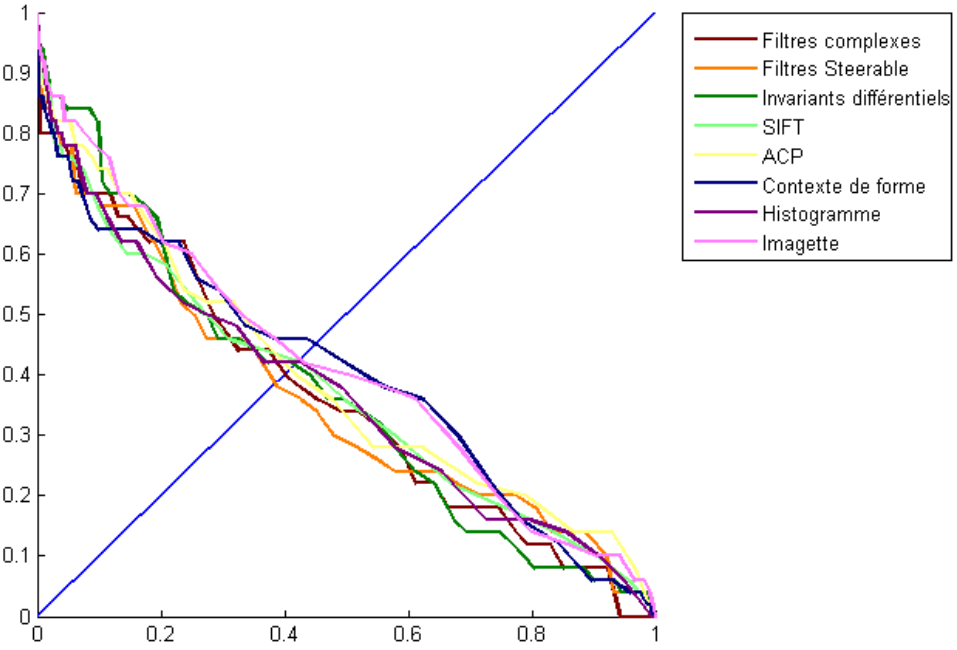
comme dans le cas de la reconnaissance de classe. Nous pouvons voir finalement que les meilleurs descripteurs sont le contexte de forme, l'ACP et SIFT.

Temps de calcul

Nous avons également calculé le temps de calcul pour chaque descripteur. Le tableau 3.16 présente le temps de calcul entre deux images et entre deux objets. Nous pouvons voir que, à l'exception du descripteur Histogramme, les descripteurs ont



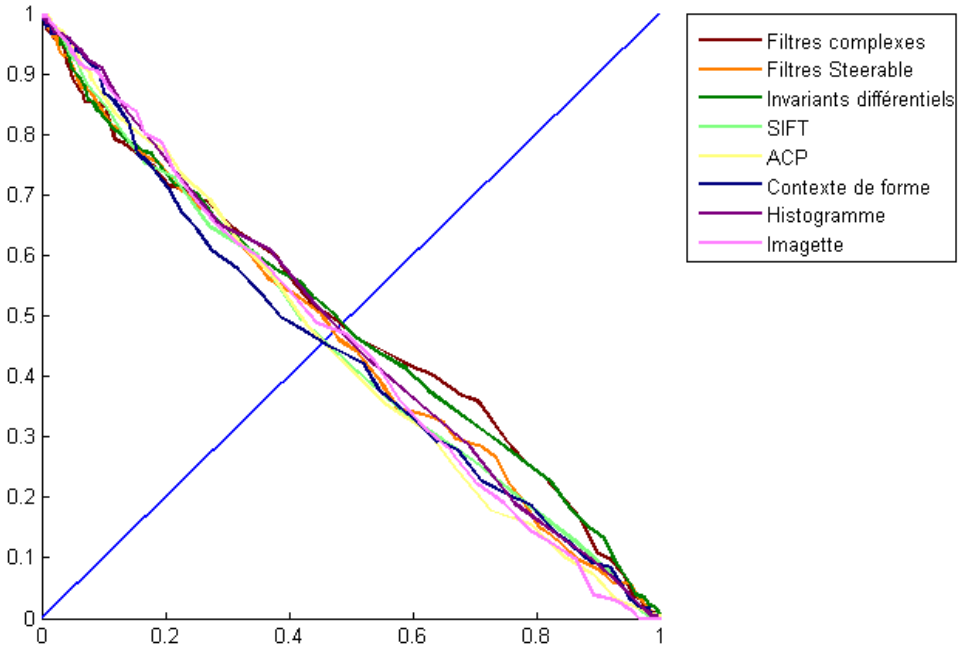
(a) nombre d'associations



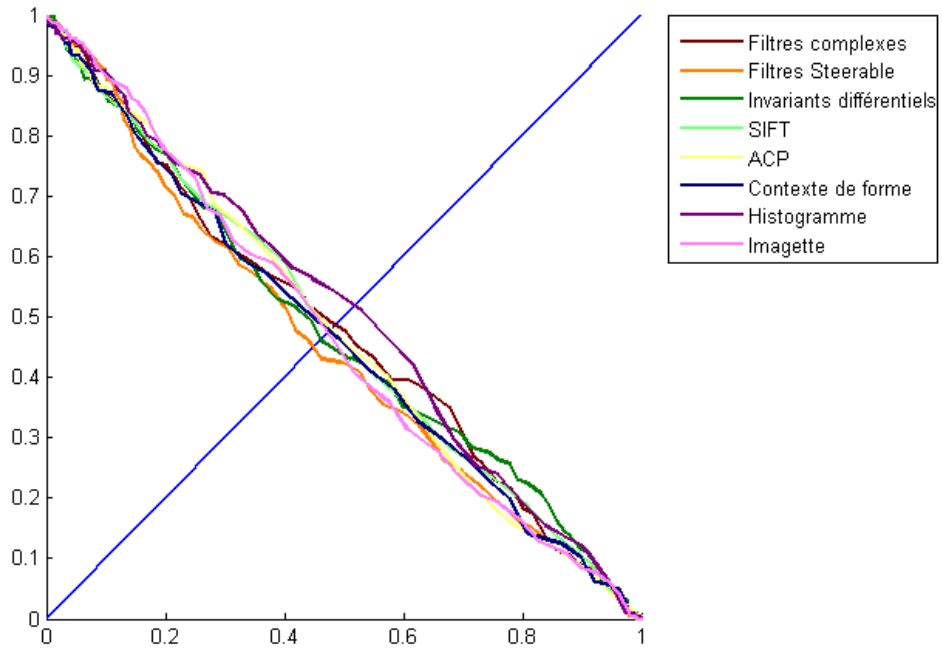
(b) Normalisation

FIGURE 3.39 – Courbes DET pour la reconnaissance de classes

environ le même temps de calcul compris entre 6 et 10 secondes pour le calcul de la distance entre deux classes.



(a) nombre d'associations



(b) Normalisation

FIGURE 3.40 – Courbes DET pour la catégorisation de classes

3.4.3 Discussions

Nous avons vu que les résultats obtenus sont relativement faibles. Néanmoins, si l'on prend en compte la difficulté de la base de données, cela permet d'expliquer ces

TABLE 3.15: Résultats pour la reconnaissance de catégories

Descripteurs	Nombre d'associations	Normalisation
Filtres complexes	48.46 %	48.47 %
Filtres Steerable	47.16 %	**45.16 %**
Invariants différentiels	48.84 %	46.21 %
SIFT	45.77 %	47.91 %
ACP	**45.72 %**	47.98 %
Contexte de forme	45.89 %	47.86 %
Histogramme	48.15 %	52.06 %
Imagette	48.21 %	47.24 %

TABLE 3.16: Temps moyen de calcul entre deux exemples et deux objets

Descripteurs	Temps de calcul entre deux exemples (ms)	Temps de calcul entre deux objets (s)
Filtres complexes	17,0	6,81
Filtres Steerable	16,4	6,54
Invariants différentiels	**14,9**	**5,97**
SIFT	24,0	9,61
ACP	20,9	8,36
Contexte de forme	20,7	8,26
Histogramme	49,5	19,81
Imagette	19,3	7,70

résultats. En effet, les performances attendues était d'environ 45% à 55% d'erreur. Avec notre méthode, la performance en reconnaissance se situe entre 33% et 42% d'erreur, soit un réel gain de performance. L'objet de cette étude n'était pas d'améliorer ces résultats mais de déterminer la plus efficace.

La performance en catégorisation est légèrement en retrait par rapport à la reconnaissance. Nous pouvons voir que la normalisation n'apporte pas un gain de performance par rapport au nombre de correspondance. Enfin, nous pouvons également voir que le descripteur SIFT ainsi que l'ACP donnent de meilleurs résultats que les autres descripteurs.

3.5 Conclusions

Ce chapitre a concerné trois études comparatives, concernant la localisation et la reconnaissance. Pour la localisation, notre étude se base sur la définition de propriétés que doit respecter une métrique pour être considérée comme pertinente. Ces propriétés, combinés avec une base de données synthétiques conséquente, nous a permis de mettre en avant plusieurs faits. Tout d'abord, les métriques en provenance

du projet Robin, prises individuellement, ne permettent pas d'évaluer correctement des résultats de localisation. Néanmoins, les trois métriques prises ensembles permettent de compenser les défauts de chacune : les métriques ROB_{cor} et ROB_{com} sont insensibles à la translation mais ROB_{loc} la gère très bien, ROB_{loc} ne gère pas la mise à l'échelle mais celle-ci est bien gérée par les autres métriques. Nous avons également vu que les métriques basées contour obtiennent de moins bonnes performances que les métriques basées masque. De plus, parmi les métriques basées contour, celles de type topologique se rapprochent plus des performances obtenues par les métriques basées masque. De cette étude, nous pouvons faire ressortir les métriques développées par Martin *et coll.* dans [47] ainsi que la métrique utilisée dans la compétition Pascal [22].

Pour la reconnaissance, nous avons effectué deux études comparatives. Dans la première, nous nous sommes intéressés aux modes de représentations des objets. Cela nous a permis de déterminer que les ensembles de points sont un bon compromis offrant des performances correctes et un temps de calcul raisonnable. La seconde étude que nous avons menée visait à comparer les descripteurs lors de la création de matrice de similarité de classes. Cette étude à été menée sur la base Caltech256 qui a l'avantage de bénéficier d'une taxonomie. La difficulté de cette base nous empêche d'avoir des résultats très bons. Cependant, nous avons vu que notre mesure de similarité permettait d'avoir des résultats satisfaisants en reconnaissance, et que la perte de performance lors de la catégorisation était faible. Ces deux études comparatives ont permis la définition de matrice de distance entre les classes, comblant ainsi les lacunes liées à l'utilisation de variables qualitatives pour l'identification des classes. Pour le calcul de cette matrice, nous avons vu que l'utilisation de la représentation des classes d'objets par des ensembles de points était efficace. La mesure de similarité utilisée est alors la double mise en correspondance. Concernant les descripteurs, nous avons vu que SIFT et ACP-SIFT permettent d'obtenir de bonnes performances et cela en un temps raisonnable.

Nous allons utiliser ces résultats afin de développer une méthode d'évaluation globale, prenant en compte en même temps la localisation et la reconnaissance lors de l'évaluation dans le chapitre suivant.

Chapitre 4

Proposition d'une méthode d'évaluation d'un résultat d'interprétation d'image

Ce chapitre présente la méthode d'évaluation d'un résultat d'interprétation d'image que nous avons développée. Cette méthode permet de prendre en compte, en même temps, connaissant la vérité terrain associée à un résultat, la qualité de la localisation, de la reconnaissance ainsi que de la détection des objets. Cette méthode est paramétrable, ce qui permet de l'adapter à une application particulière.

Sommaire

4.1 Introduction

Nous avons vu au cours du chapitre 2 que de nombreuses méthodes permettent de comparer un résultat de localisation avec une vérité terrain, ou bien un ensemble de résultats d'interprétation. Des compétitions existantes, telles que le Pascal VOC Challenge [22] ou la compétition française Robin [2], disposent d'une

vérité terrain aussi bien pour la localisation que pour la reconnaissance afin d'évaluer des algorithmes d'interprétation d'images. Cependant, elles se limitent à évaluer soit la localisation soit la reconnaissance, mais jamais les deux en même temps. De plus, ces compétitions, ainsi que les courbes ROC ou Précision/Rappel, évaluent les algorithmes sur une base de données. Il y a donc un manque concernant l'évaluation d'un seul résultat d'interprétation et non pas l'évaluation d'un algorithme sur une base de données.

Nous avons travaillé sur une mesure de performance qui, à partir d'une vérité terrain, permettrait d'évaluer en même temps la localisation, la reconnaissance et la détection des objets interprétés dans une scène. Par exemple, nous pouvons voir à la figure 4.1 une image avec quatre résultats d'interprétation possibles. Sur ces exemples, nous souhaitons automatiquement déterminer le meilleur résultat d'interprétation. L'objectif est d'obtenir une méthode adaptable en fonction de l'application visée (importance de la localisation par rapport à la reconnaissance) et pouvant être aussi bien utilisée avec les algorithmes qui fournissent une notion de confiance pour chaque objet détecté qu'avec ceux qui n'en fournissent pas.

4.2 Méthode développée

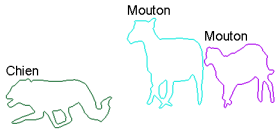
La méthode que nous avons développée est composée de quatre étapes, comme nous pouvons le voir à la figure 4.2 :

- Une mise en correspondance,
- Une évaluation locale pour chaque objet mis en correspondance,
- Une prise en compte de la sous et de la sur-détection,
- Un calcul du score global.

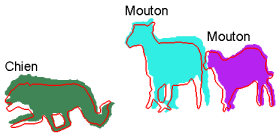
4.2.1 La mise en correspondance

La mise en correspondance des objets permet de savoir à quel objet de la vérité terrain correspondent les objets détectés par l'algorithme. Cette étape est nécessaire pour les deux étapes suivantes : le calcul des scores locaux et la compensation de la sous et sur-détection. En effet, les objets sur-détectés correspondent à des objets présents dans le résultat d'interprétation qui ne correspondent à aucun objet de la vérité terrain. Concernant les objets sous-détectés, ce sont les objets présents dans la vérité terrain mais qui sont absents du résultat d'interprétation.

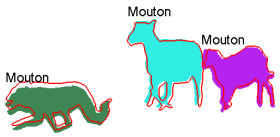
(a) Image originale



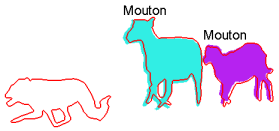
(b) Vérité Terrain



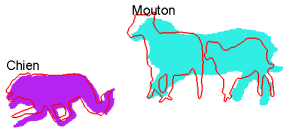
(c) Résultat 1



(d) Résultat 2



(e) Résultat 3



(f) Résultat 4

FIGURE 4.1 – Exemples de résultats d'interprétation sur une scène (Image originale tirée de [22]) : sur les résultats, le contour rouge présente la vérité terrain de la localisation

Afin de faire cette mise en correspondance, nous calculons une matrice de recouvrement, de manière similaire aux travaux présentés dans [75]. Chaque ligne de la matrice correspond à un objet présent dans la vérité terrain, tandis que les colonnes correspondent aux objets présents dans le résultat d'interprétation. Dans la cellule (u, v), nous indiquons le recouvrement de l'objet u de la vérité terrain et de l'objet v du résultat d'interprétation. Le recouvrement est calculé avec la métrique PAS que nous avons étudiée dans le chapitre précédent :

$$PAS(I_{vt}, I_i) = \frac{\text{Card}(I_{vt}^{Re} \cap I_i^{Re})}{\text{Card}(I_{vt}^{Re} \cup I_i^{Re})} \tag{4.1}$$

avec $\text{Card}(I_{vt}^{Re})$ le nombre de pixels dans l'objet de la vérité terrain, et $\text{Card}(I_i^{Re})$ le nombre de pixels présents dans l'objet détecté par l'algorithme d'interprétation. Le score de recouvrement est compris dans $[0, 1]$, 1 étant le score optimal. La figure 4.3

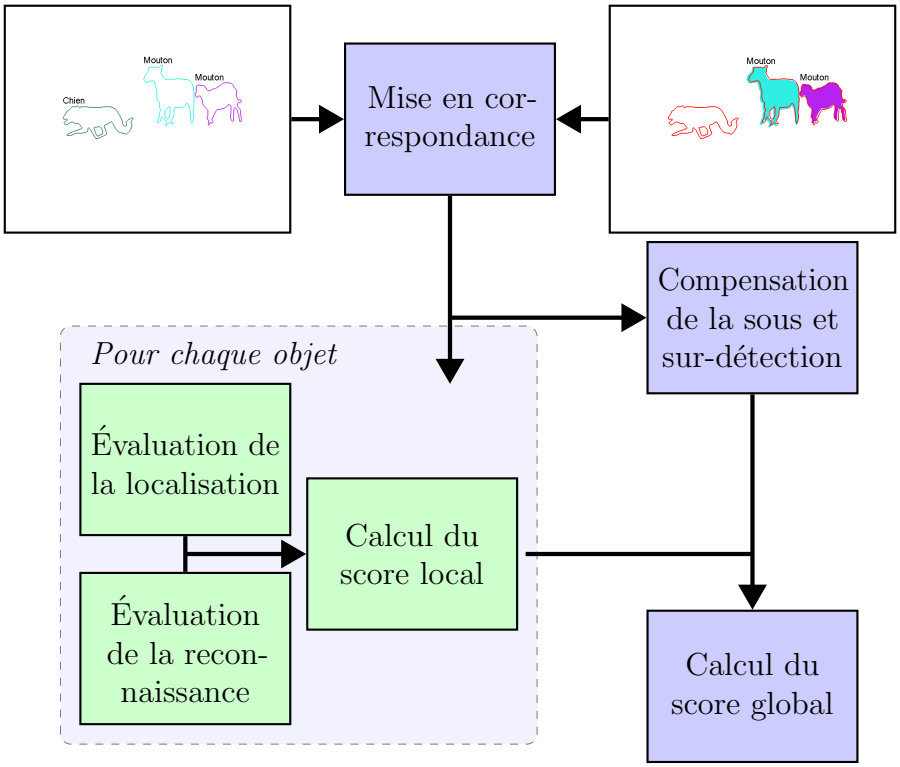


FIGURE 4.2 – Schéma de l'évaluation globale d'un résultat d'interprétation

présente l'ensemble $I_{vt}^{Re} \cap I_i^{Re}$ en vert tandis que l'ensemble $I_{vt}^{Re} \cup I_i^{Re}$ est la réunion des régions verte, bleue et rouge.

(a) Vérité terrain (b) Résultat de localisation (c) Superposition des deux régions

FIGURE 4.3 – La métrique PAS correspond à l'ensemble vert de pixels en commun divisé par l'ensemble des pixels localisés, c'est-à-dire vert, rouge et bleu

A partir de cette matrice, nous avons choisi d'implémenter deux méthodes pour la mise en correspondance. La première méthode est en « un pour un » et consiste à assigner un seul objet de la vérité terrain à un seul objet du résultat d'interprétation. Cette méthode est également utilisée dans [22]. Nous utilisons l'algorithme hongrois

[53] (voir section 2.4.2) afin de calculer cette mise en correspondance.

La deuxième méthode, dite « multiple », permet que chaque objet de la scène puisse être assigné à plusieurs objets dans la vérité terrain ou le résultat d'interprétation. Pour cela, nous faisons simplement un seuillage sur la matrice de recouvrement. Cela permet de prendre en compte, par exemple, le fait qu'un groupe d'objets soit reconnu comme un seul objet par l'algorithme. Ceci est utilisé dans [75, 46].

Après cette mise en correspondance, nous obtenons une matrice de correspondance, où un 1 dans une cellule (u, v) indique que l'objet u de la vérité terrain est mis en correspondance avec l'objet v du résultat d'interprétation. Par défaut, la méthode utilisée est la méthode « multiple » avec un seuil de $0,2$. Ces paramètres sont modifiables par l'utilisateur afin d'adapter la méthode d'évaluation en fonction de l'application visée.

0,65				
			0,76	
	0,21	0,72		

(a) Matrice de recouvrement

1				
			1	
		1		

(b) Matrice de correspondance (méthode « un pour un »)

1				
			1	
	1	1		

(c) Matrice de correspondance (méthode « multiple »)

FIGURE 4.4 – Exemples de matrices de recouvrement et de correspondance

4.2.2 Le calcul du score local

Pour chaque objet mis en correspondance, c'est-à-dire les cases (u, v) contenant un 1 dans la matrice de correspondance, nous calculons un score. Ce score est dit « local » car il correspond à l'évaluation d'un seul objet dans une scène en contenant potentiellement plusieurs. Il est calculé à partir de la qualité de la localisation et de la reconnaissance de l'objet. Le score de localisation S_{loc} est une version adaptée pour un objet des métriques de Martin [47] qui ont montré leurs bonnes performances dans le chapitre précédent :

$$S_{loc}(I_{vt}, I_i, u, v) = \min\left(\frac{\text{Card}(I_{vt\setminus i}^{Re(u)})}{\text{Card}(I_{vt}^{Re(u)})}, \frac{\text{Card}(I_{i\setminus vt}^{Re(v)})}{\text{Card}(I_{i}^{Re(v)})}\right) \tag{4.2}$$

avec $\text{Card}(I_{vt}^{Re(u)})$ le nombre de pixels de l'objet u présents dans la vérité terrain et $\text{Card}(I_{vt\setminus i}^{Re(u)})$ le nombre de pixels de l'objet u présents dans la vérité terrain, mais pas dans le résultat d'interprétation. Ce score est compris entre 0 et 1, 0 indiquant un recouvrement parfait des deux objets, donc une parfaite localisation.

Nous calculons ensuite un score pour la qualité de la reconnaissance. Pour cela, nous utilisons une matrice de similarité vue dans le chapitre précédent. L'utilisateur doit fournir une matrice de similarité qu'il aura précédemment calculée ou, éventuellement, construite manuellement. Cette matrice doit être calculé en fonction de la base de données utilisée. Si l'utilisateur ne fournit pas de matrice de similarité, une matrice carré est construite par défaut telle qu'elle ne contienne que des 1 avec des 0 sur la diagonale (soit la matrice $(1 - \delta_{i,j})$, avec $\delta_{i,j}$ le symbole de Kronecker), avec autant de lignes et de colonnes qu'il y a de classes dans la base de données. De plus, l'algorithme d'interprétation d'images peut fournir un indice de confiance pour chaque objet détecté. Le score est alors le suivant :

$$S_{rec}(u, v, \mu) = MS(cl(u), cl(v)) * ind(cl(u), cl(v), \mu) \tag{4.3}$$

avec

$$ind(cl(u), cl(v), \mu) = \begin{cases} \frac{1-\mu}{2} & \text{si } cl(u) = cl(v) \\ \frac{1+\mu}{2} & \text{sinon} \end{cases} \tag{4.4}$$

et μ l'indice de confiance accordé au résultat de reconnaissance, MS la matrice de similarité et $cl(u)$ la classe de l'objet u.

Suite à cela, nous calculons un score d'interprétation, qui est la combinaison de ces deux scores :

$$S(u, v) = \alpha * S_{loc}(I_{vt}, I_i, u, v) + (1 - \alpha) * S_{rec}(I_{vt}, I_i, u, v) \tag{4.5}$$

Par défaut, la valeur du paramètre α est de $0,8$, mais l'utilisateur peut choisir de le modifier afin de donner plus de poids à la reconnaissance ou à la localisation. Nous avons choisi la valeur de $0,8$ afin de prendre davantage en considération la localisation que la reconnaissance étant donnée que la pénalisation par défaut de la reconnaissance est plus importante que celle de la localisation. Après avoir calculé

ces scores, nous obtenons une matrice de scores locaux que l'on peut voir à la figure 4.5.

0,1				
			0,16	
	0,63	0,21		

(a) Matrice de scores de localisation

0				
			1	
	0	0		

(b) Matrice de scores de reconnaissance

0,08				
			0,58	
	0,5	0,17		

(c) Matrice de scores locaux

FIGURE 4.5 – Exemples de matrices de scores

4.2.3 La compensation

Après avoir calculé un score pour chaque objet mis en correspondance, nous regardons les objets sous et sur-détectés. La sous-détection correspond à des lignes de la matrice de correspondance qui n'ont pas été associées à un objet du résultat d'interprétation, donc n'ayant pas de 1. De même, les objets sur-détectés vont correspondre à des colonnes de la matrice de correspondance qui n'ont été associées à aucun objet de la vérité terrain.

Nous commençons par prendre en compte la sous-détection. Pour cela, on recherche la première ligne u sans association, puis pour cette ligne, on recherche la première colonne v sans association. On associe alors l'objet u à l'objet v dans la matrice de correspondance. Dans la matrice des scores locaux, on met alors le score 1 pour cette association, 1 correspondant à un mauvais score d'interprétation. On recommence ensuite jusqu'à ce que toutes les lignes soient associées. Pour la sur-détection, on fait le même travail en échangeant les lignes et les colonnes. On peut voir le résultat de la compensation à la figure 4.6.

4.2.4 Le calcul de score global

Le score global est calculé à partir de la matrice des scores locaux, le score global étant alors moyenne des scores locaux.

1				
			1	
	1	1		
				1

(a) Matrice de correspondance avec compensation

0,08				
			0,58	
	0,5	0,17		
				1

(b) Matrice de scores locaux avec compensation

FIGURE 4.6 – Exemples de compensation

4.2.5 Illustration

Nous avons appliqué notre méthode d'évaluation, avec les paramètres par défaut, à un résultat d'interprétation de l'image présentée à la figure 4.7.

FIGURE 4.7 – Une image originale de la base Pascal VOC challenge 2007

À la figure 4.8, nous pouvons voir chaque étape de l'évaluation. La vérité terrain présente sept objets : les quatre premiers sont de la classe « personne », suivis d'un objet de la classe « bus », puis « avion » et enfin un objet de la classe « voiture » qui est peu visible. Le résultat d'interprétation présente quatre objets : le premier de la classe « camion », suivi d'un objet de la classe « avion » puis de deux objets de la classe « personne ».

Nous pouvons constater que l'avion et le bus ont été bien localisés, bien que le bus soit reconnu en tant que camion. Les quatre personnes ont bien été reconnues mais mal localisées. En effet, un seul objet a été détecté à la place de trois. Enfin, la

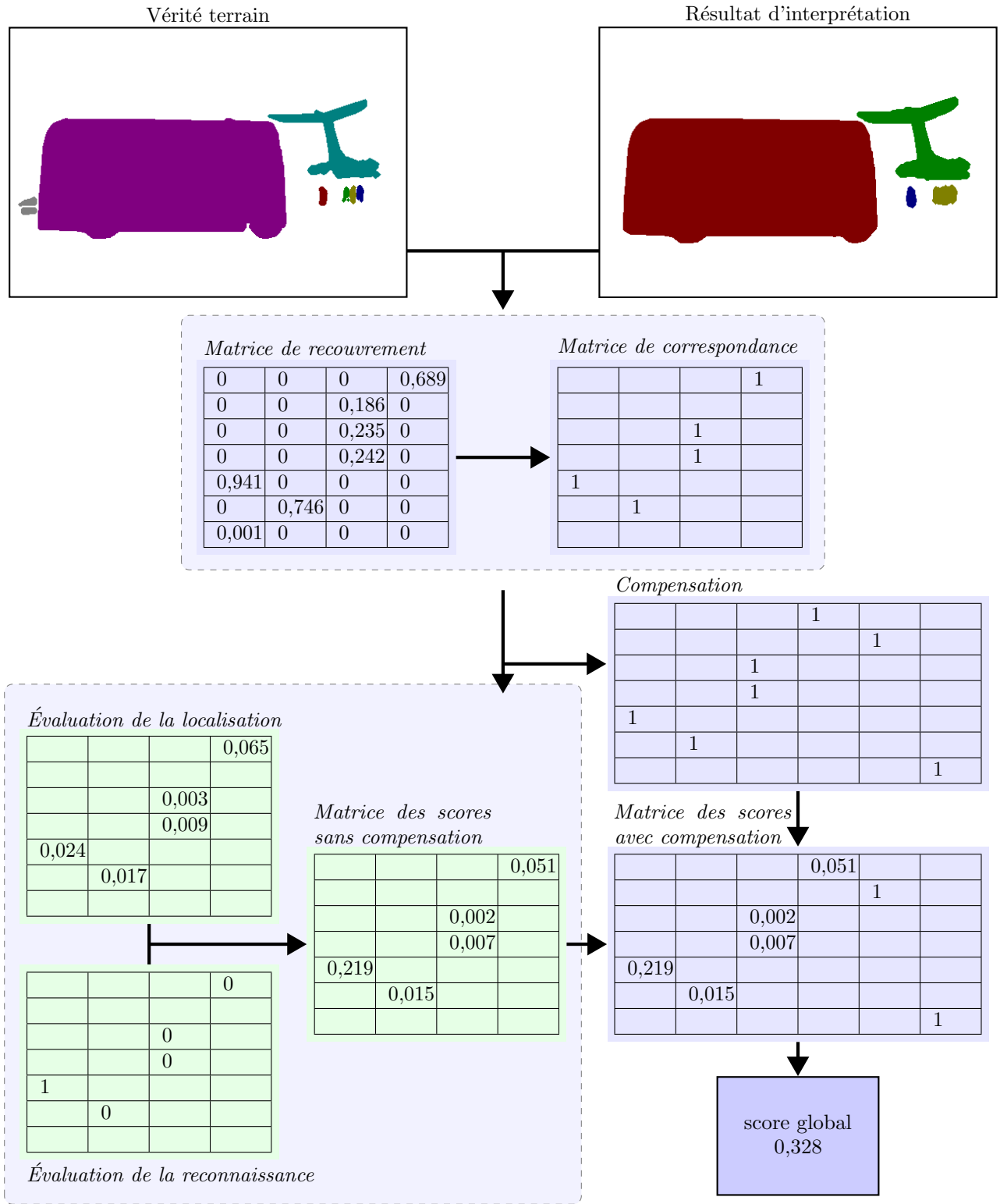


FIGURE 4.8 – Exemple d'évaluation globale d'un résultat d'interprétation

voiture n'a pas été détectée du tout.

La première étape consiste à mettre en correspondance les objets de la vérité terrain et ceux du résultat d'interprétation. La matrice de recouvrement présente le résultat de la métrique PAS pour chaque couple (u, v) d'objets. Ainsi, pour le bus, qui est l'objet 5 de la vérité terrain et l'objet 1 du résultat d'interprétation, le recouvrement des masques est important. Cela amène un score de recouvrement de 0,941. Ce score étant supérieur au seuil, ces deux masques sont mis en correspondance comme on peut le voir dans la matrice de correspondance. Il en va de même pour l'avion ainsi qu'une personne qui sont très clairement mis en correspondance. Le groupe de trois personnes correspond aux objets 2, 3 et 4 de la vérité terrain et à l'objet 3 du résultat de reconnaissance. Le score de recouvrement est donc moins important avec 0,235 et 0,242 pour 2 objets et 0,186 pour le dernier. Le seuil étant de 0,2, seulement 2 objets de la vérité terrain sont mis en correspondance avec l'objet du résultat d'interprétation.

La seconde étape consiste à calculer des scores locaux, c'est-à-dire pour chaque objet mis en correspondance. Pour cela, une matrice de scores de localisation est calculée, ainsi qu'une matrice des scores de reconnaissance. Pour la localisation, on peut voir que le groupe d'individus est bien localisé avec des scores de localisation inférieurs à 0,01 tandis que l'autre individu est le moins bien localisé avec un score de 0,065, ce qui reste un bon score de localisation. L'avion et le bus sont bien localisés avec des scores de localisation de 0,017 et 0,024. Pour la reconnaissance, nous pouvons voir qu'à l'exception du bus, les objets sont bien reconnus ce qui explique qu'ils aient un score de 0. Le bus étant reconnu comme un camion, son score est de 1. L'utilisation de matrice de confiance permettrait de réduire ce score étant donné que ces deux objets sont assez similaires. Une matrice de scores locaux est ensuite calculée comme la combinaison des deux matrices précédentes. Nous pouvons voir que le score du bus est fortement impacté par la mauvaise reconnaissance.

La troisième étape est la compensation. On part pour cela de la matrice de correspondance afin d'identifier les lignes et/ou les colonnes n'ayant pas été affectées. Nous pouvons voir que toutes les colonnes contiennent au moins un 1, ce qui signifie que tous les objets du résultat d'interprétation ont été affectés à au moins un objet de la vérité terrain, et qu'il n'y a pas de sur-détection. Cependant, les lignes 2 et 7 sont vides, les objets correspondant dans la vérité terrain n'ont pas été convenablement détectés. Cette sous-détection est alors compensée en ajoutant des

scores 1 dans les lignes correspondantes. Des colonnes sont ajoutées afin de ne pas affecter ces objets à des objets du résultat d'interprétation déjà correctement détectés.

La dernière étape consiste à calculer la matrice des scores locaux en prenant en compte la compensation. Cette matrice présente les scores locaux calculés précédemment en ajoutant les scores provenant de la compensation. A partir de cette matrice, on calcule le score moyen ce qui nous donne le score global. Dans notre cas, on calcule la moyenne de 7 scores obtenus : les 5 provenant de la mise en correspondance et les 2 provenant de la compensation. Le score final obtenu est de 0,328. Ce score est assez élevé car l'absence de détection de deux objets est très pénalisante : les objets manquants contribuent à hauteur de 0,285 et les objets présents à hauteur de 0,043.

4.3 Validation de la méthode

4.3.1 Protocole expérimental

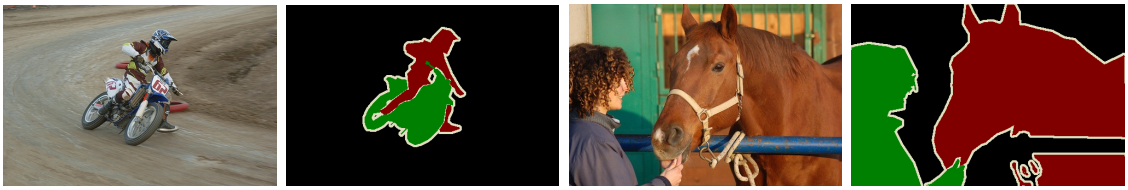
Nous avons testé notre méthode d'évaluation globale sur une grande base de données extraite de la base de données Pascal VOC challenge 2008. Cette base nous a fourni un ensemble de vérités terrain intéressantes contenant à la fois la localisation par des masques ainsi qu'une classe associée à chaque objet. Nous avons ensuite appliqué sur les objets issus de ces vérités terrain des altérations, puis nous avons étudié le comportement de notre méthode en fonction des différentes altérations. La suite de cette section présente la base de données ainsi que les altérations que nous avons appliquées aux vérités terrain.

Base de données

Nous avons utilisé la base de données du Pascal VOC challenge 2008 [1] [22]. Parmi cette base de données, plusieurs ensembles sont disponibles, chacun correspondant à un type d'algorithme. Nous avons utilisé l'ensemble « Segmentation Taster Set ». Cet ensemble d'images présente des vérités terrain dont la localisation est un masque, ce qui correspond à notre algorithme d'évaluation. Nous pouvons voir quelques exemples d'images issues de cet ensemble de la base de données à la figure 4.9.

Dans la table 4.1, nous référençons le nombre d'images contenant un nombre N d'objets. Nous pouvons voir qu'une majorité des images contient seulement un ou deux objets. Sur les 1022 images disponibles dans la base, 1002 ont entre 1 et 8

1. http://www.pascal-network.org/challenges/VOC/databases.html



(a) Image (b) Vérité terrain (c) Image (d) Vérité terrain

FIGURE 4.9 – Images provenant de l'ensemble « Segmentation Taster Set »

objets. Les 20 images restantes ont entre 9 et 21 objets. Comme il n'y avait pas au minimum 10 images contenant N (N>8) objets, ces images ont été rejetées. En tout, cela représente un total de *2 134* objets que nous avons altérés.

Nombre d'objets	1	2	3	4	5	6	7	8
Nombre d'images	498	237	106	61	40	32	16	12

TABLE 4.1: Nombre d'images

La base de données contient en tout 20 classes différentes : « avion », « bicyclette », « oiseau », « bateau », « bouteille », « bus », « voiture », « chat », « chaise », « vache », « table à manger », « chien », « cheval », « moto », « personne », « plante en pot », « mouton », « sofa », « train » et « télévision ». Nous avons ordonné ces classes selon les catégories de la base Caltech256 [1] (voir figure 4.10).

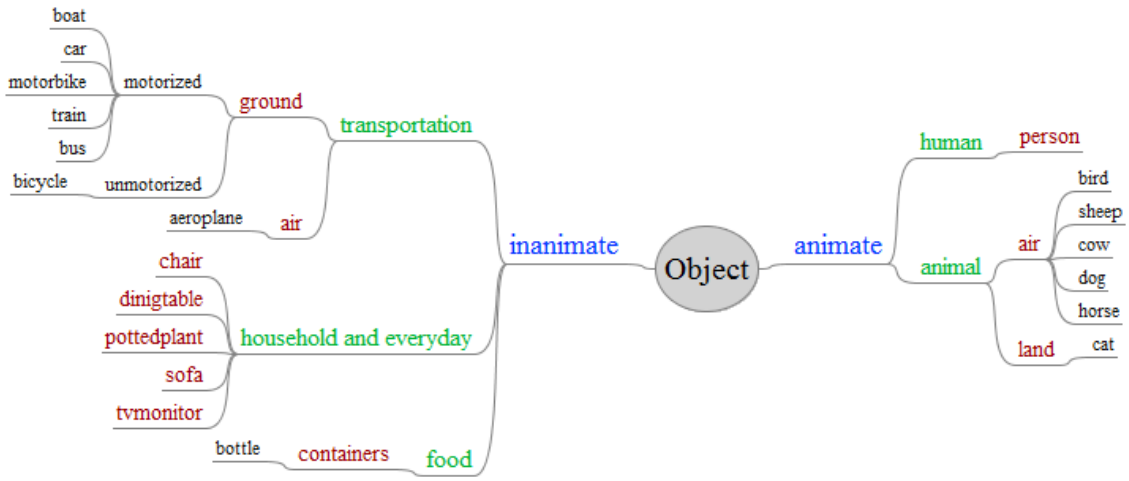


FIGURE 4.10 – Classes de la base Pascal 2008 représentées en utilisant la taxonomie de la base Caltech256

Altérations

Tout d'abord, nous avons considéré les mêmes altérations que lors de l'étude comparative de la localisation : la translation, la mise à l'échelle, la rotation et enfin la perspective. Pour chacune de ces altérations, nous avons utilisé un paramètre d'altération allant de 1 à 20 dans deux directions différentes : horizontale et verticale pour les altérations de translation, mise à l'échelle et perspective, sens horaire et anti-horaire pour la rotation. Cela nous amène à 160 altérations pour chaque objet, soit un total de *341 440* altérations considérées.

Nous avons ensuite considéré des altérations de la reconnaissance en altérant la classe des objets. Pour cela, nous avons remplacé la classe de 1 à tous les objets présents dans l'image par la classe « *Autre* ». De même, nous avons regardé l'effet de la sous et sur-détection d'objets sur le comportement de notre méthode d'évaluation. Pour cela, nous avons ajouté ou supprimé de 1 à 8 objets, de sorte qu'il ne soit en correspondance avec aucun autre objet présent dans l'image.

Paramétrage

Nous nous sommes également intéressés à l'évolution de la métrique en fonction des différents paramètres de celle-ci. Nous avons alors regardé tout d'abord l'effet du choix de mise en correspondance ainsi que l'effet du seuil sur la méthode de mise en correspondance « multiple ». Nous avons donc comparé la mise en correspondance « un pour un » et « multiple », avec un seuil valant 0,2, 0,3, 0,4 et 0,5.

Nous avons également regardé les effets de l'utilisation d'un indice de confiance pour la reconnaissance, ainsi que l'utilisation d'une matrice des distances entre les différentes classes d'objets.

4.3.2 Résultats expérimentaux

Localisation

Les figures 4.11 à 4.14 présentent l'évolution moyenne de la métrique globale en fonction de différentes altérations de la localisation et selon le nombre d'objets présents dans la vérité terrain. Chaque courbe présente l'évolution de la métrique en fonction de la puissance d'altération de la localisation. La métrique est ici présentée avec les valeurs par défaut. Nous remarquons tout d'abord que plus le nombre d'objets dans la vérité terrain augmente, et moins le critère est pénalisant. Ceci est normal puisque le score global est la moyenne des scores locaux. Ce résultat est donc correct

et en adéquation avec la manière dont nous avons développé cette méthode globale. Nous pouvons voir également que les propriétés définies dans la section 3.2.1 sont respectées. Quelle que soit l'altération considérée ou le nombre d'objets, les courbes sont uniformément régulières et strictement monotones. Nous pouvons voir que la métrique a bien la propriété de séparabilité également. De plus, bien que cela ne soit pas visible sur les courbes, la métrique est symétrique.

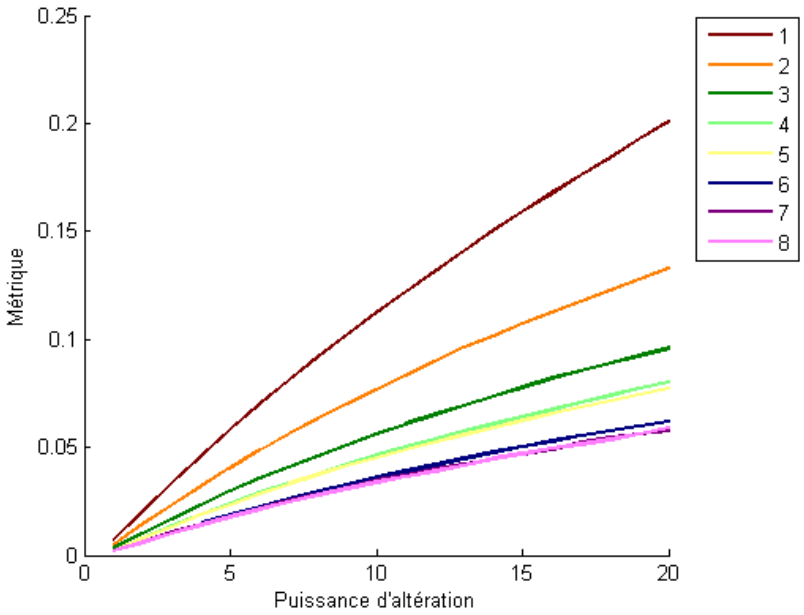


FIGURE 4.11 – Résultats concernant la localisation - Translation

Nous pouvons également remarquer que la translation et la rotation sont les deux altérations les plus pénalisées (à puissance d'altération égale), suivies par le changement d'échelle puis le changement de perspective. Cela semble correct au regard de la figure 4.15, où toutes les images ont été altérées avec le même paramètre de 20. La métrique évolue donc correctement en ce qui concerne la localisation.

Reconnaissance

Concernant la reconnaissance, nous avons étudié l'évolution de la métrique en fonction du nombre d'objets dont nous avons altéré la classe. Étant donné que nous travaillons avec les paramètres par défaut, la classe affectée à l'objet altéré n'a pas d'importance sur le résultat. Nous pouvons voir à la figure 4.16 l'évolution de la métrique globale en fonction du nombre d'objets altérés, les différentes courbes

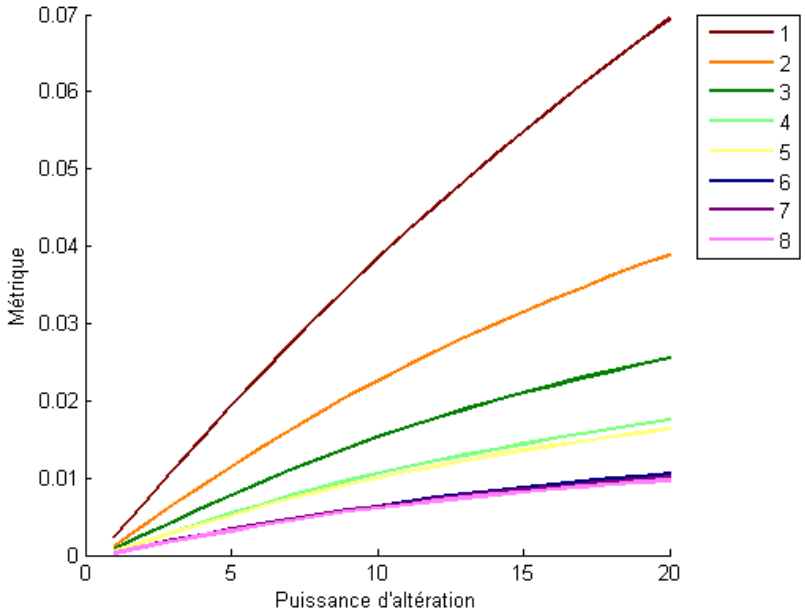


FIGURE 4.12 – Résultats concernant la localisation - Mise à l'échelle

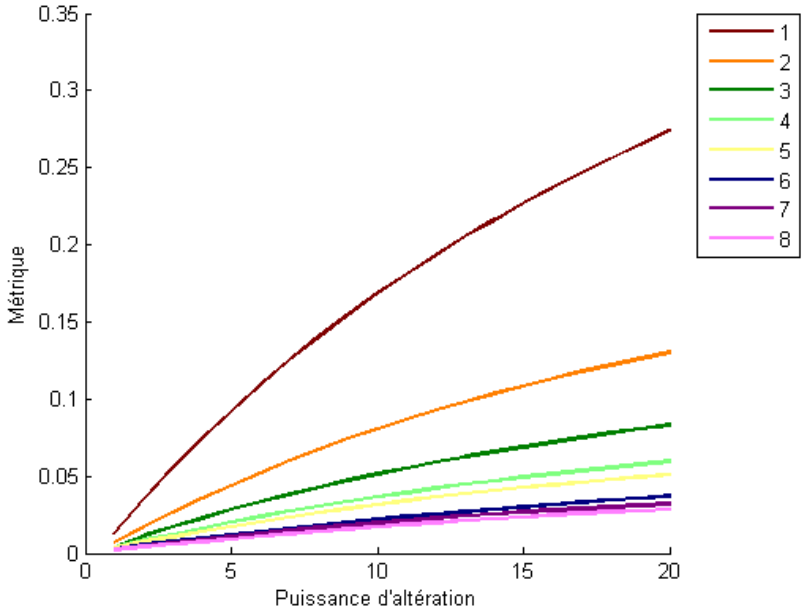


FIGURE 4.13 – Résultats concernant la localisation - Rotation

représentant le nombre d'objets dans la vérité terrain.

Nous pouvons remarquer que la métrique réagit bien. La pénalisation maximale, d'une valeur de 0,2 ($1 - \alpha$, le paramètre α étant fixé par défaut à 0,8), est atteinte

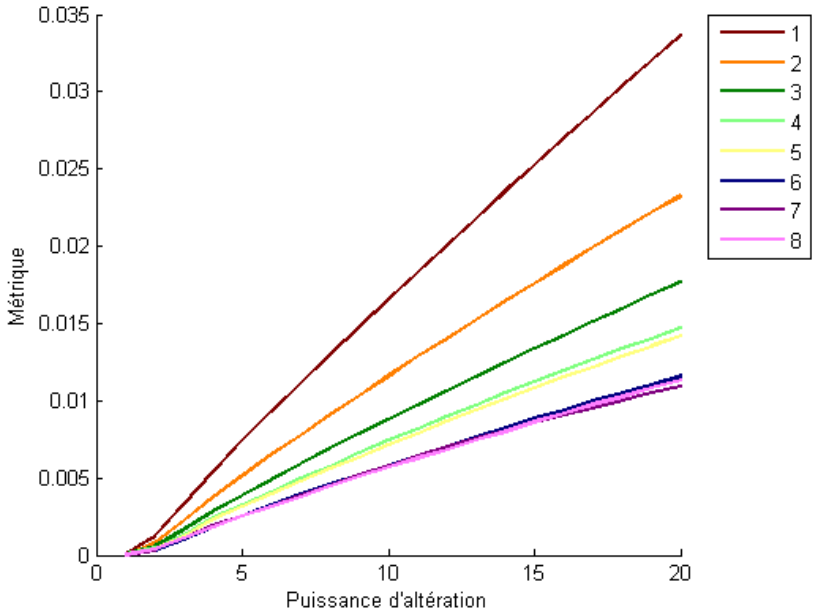
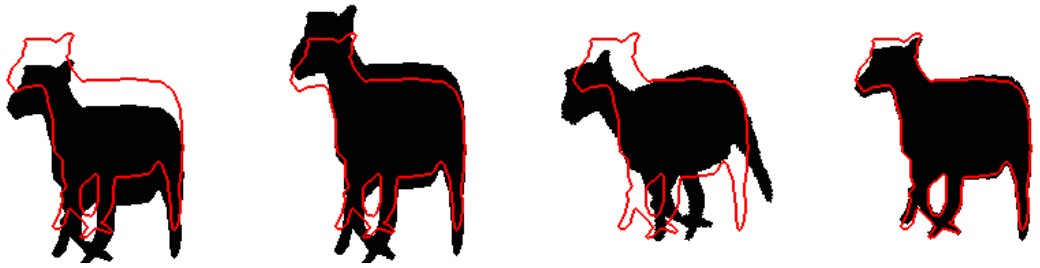


FIGURE 4.14 – Résultats concernant la localisation - Perspective



(a) Translation (b) Mise à l'échelle (c) Rotation (d) Perspective

FIGURE 4.15 – Quatre images avec la même puissance d'altération

lorsque tous les objets de la vérité terrain ont été altérés.

Sous et sur-détection

Nous avons ensuite étudié l'effet de la sous et sur-détection sur l'évolution de la métrique globale. Nous avons étudié l'évolution de la métrique d'évaluation, avec les paramètres par défaut, en fonction du nombre d'objets supprimés de la vérité terrain ou bien ajoutés. La figure 4.17 présente l'évolution de la métrique en fonction du nombre d'objets altérés (les nombres négatifs indiquent les objets supprimés et les positifs les objets ajoutés) où chaque courbe correspond à un nombre d'objets dans la vérité terrain.

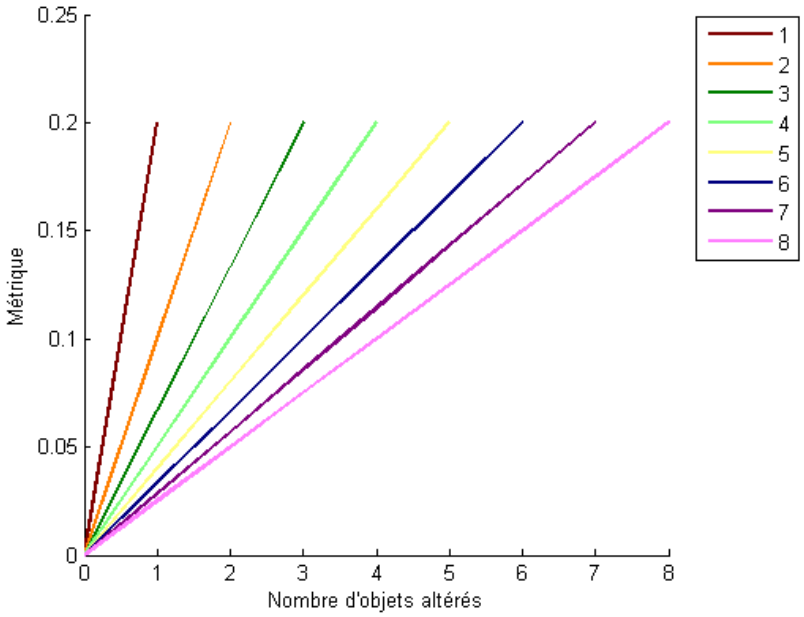


FIGURE 4.16 – Résultats concernant la reconnaissance

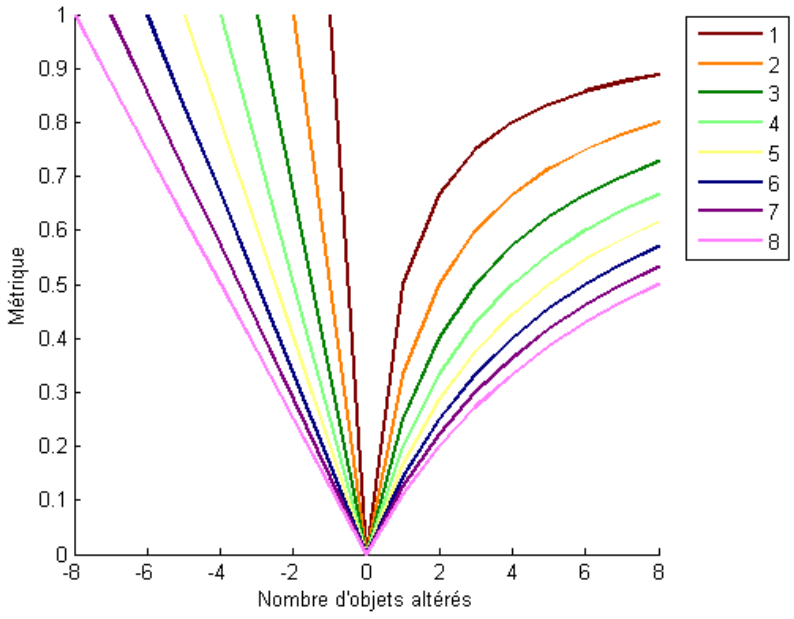


FIGURE 4.17 – Résultats concernant la sous et sur-détection

Nous pouvons voir que ces situations sont correctement gérées et que la métrique est toujours plus pénalisante lorsque le nombre d'objets dans la vérité terrain est

faible. Nous pouvons noter que la sous-détection est légèrement plus pénalisée que la sur-détection.

Tout cela montre que la métrique, lorsqu'elle est utilisée avec les critères par défaut, donne de bons résultats. Cependant, il peut être intéressant de le paramétrer selon une application visée. La suite de cette section présente l'effet du paramétrage sur le comportement de cette métrique.

Paramétrage

Mode de mise en correspondance

Le premier paramètre que nous avons considéré est le mode de mise en correspondance, ainsi que le seuil de mise en correspondance. Pour cela, nous nous sommes intéressés à l'évolution de notre métrique d'évaluation en fonction des quatre altérations de la localisation. Nous nous sommes limités à l'étude des vérités terrain contenant un seul objet. Chaque courbe de la figure 4.18 représente un paramétrage différent de notre méthode d'évaluation.

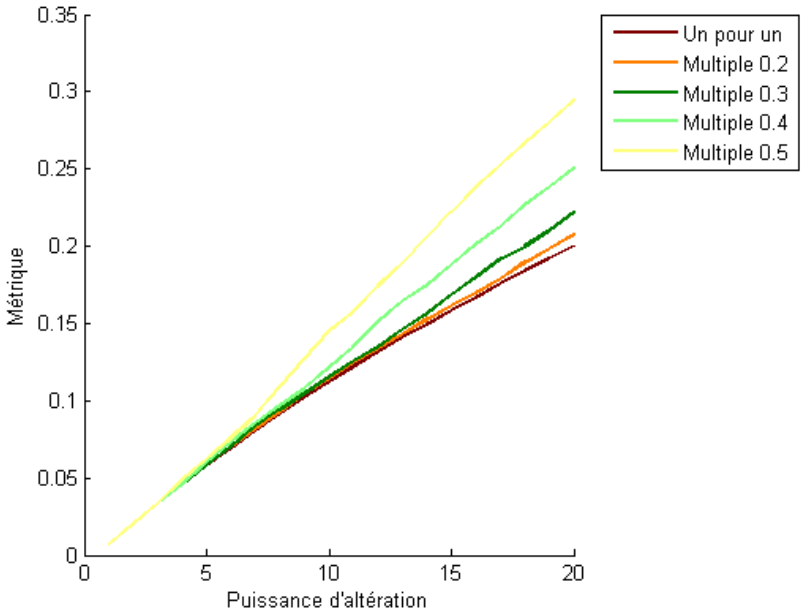


FIGURE 4.18 – Résultats concernant l'effet du paramétrage sur la localisation - Translation

Nous pouvons remarquer sur les résultats obtenus que le paramétrage « multiple » est plus pénalisant que le paramétrage « un pour un ». De plus, plus le seuil est élevé,

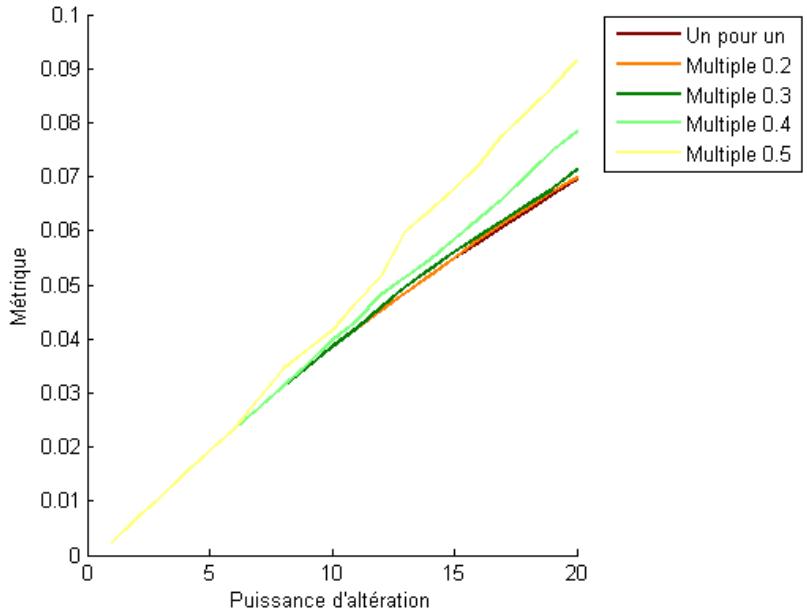


FIGURE 4.19 – Résultats concernant l'effet du paramétrage sur la localisation - Mise à l'échelle

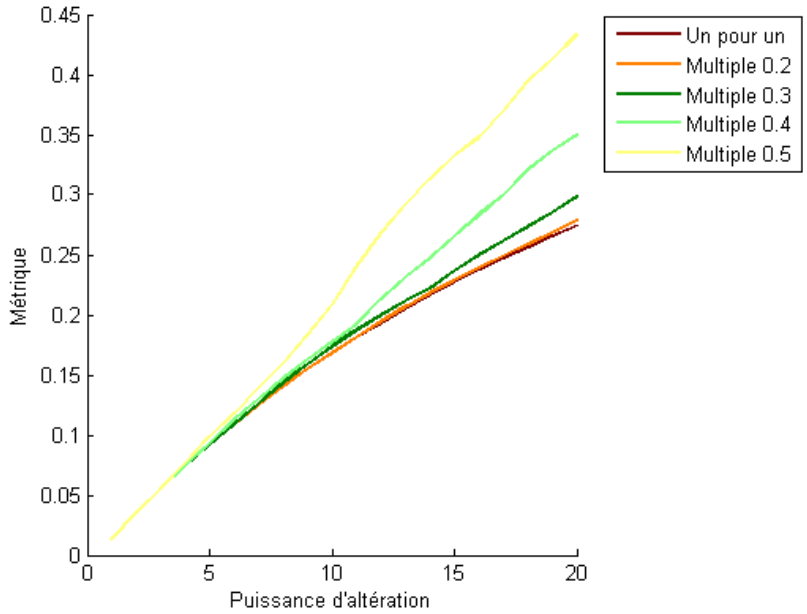


FIGURE 4.20 – Résultats concernant l'effet du paramétrage sur la localisation - Rotation

et plus les altérations sont pénalisées. Cela s'explique par le fait qu'il y ait un seul objet par vérité terrain. Ainsi, la méthode « un pour un » associe toujours l'objet

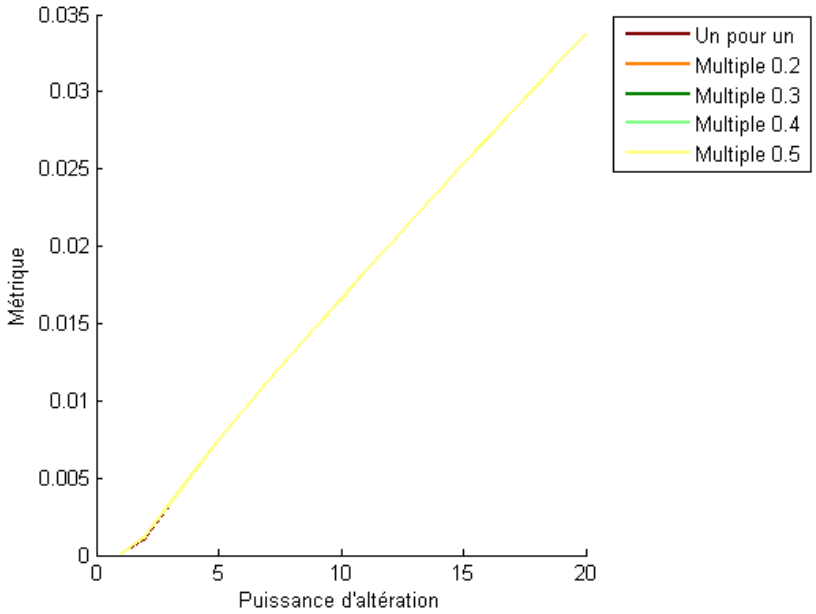


FIGURE 4.21 – Résultats concernant l'effet du paramétrage sur la localisation - Perspective

altéré à l'objet de la vérité terrain, tandis que la méthode « multiple » ne l'associera plus à partir d'une certaine altération, et cela se produira d'autant plus rapidement que le seuil est élevé. Notons également que cela se produit plus rapidement avec les altérations de rotation et de translation qu'avec l'altération de mise à l'échelle. Enfin, l'altération de perspective n'est pas assez importante pour que cela se produise.

Indice de confiance

Nous avons ensuite regardé l'effet de l'indice de confiance sur le résultat de reconnaissance. Pour cela, la figure 4.22 présente la valeur de la variable *ind* en fonction de l'indice de confiance μ, la courbe verte dans le cas où la classe est bien reconnue, et rouge dans le cas d'une erreur.

Nous voyons clairement que plus la confiance accordée est importante, plus le score de reconnaissance est impacté car la différence entre la courbe rouge et la courbe verte augmente. Dans le cas d'une bonne reconnaissance, une augmentation de la confiance permet de diminuer la valeur de l'indice multiplicateur et donc du score de reconnaissance. Le score de reconnaissance étant plus bas, le score global le devient également, signe d'une meilleure interprétation. Au contraire, dans le cas

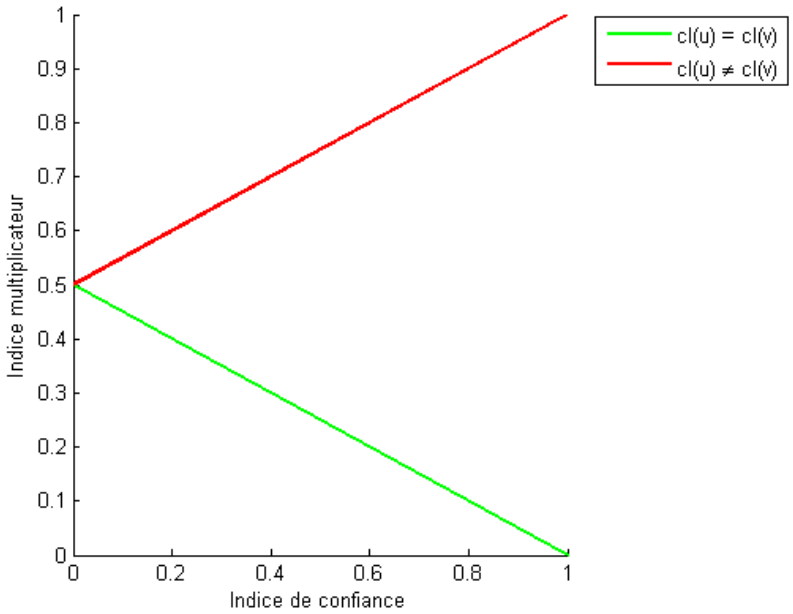


FIGURE 4.22 – Valeur de l'indice multiplicateur en fonction de l'indice de confiance

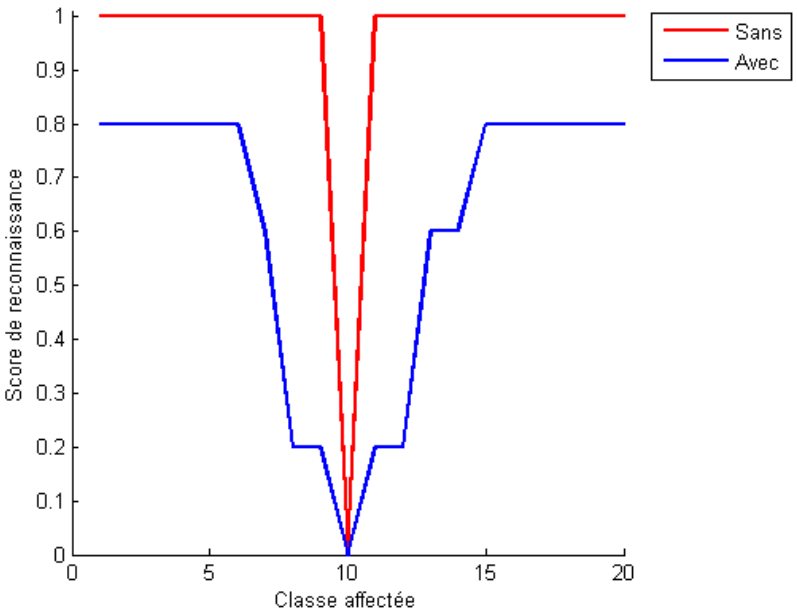
d'une mauvaise reconnaissance, l'indice multiplicateur augmente avec la valeur de l'indice de confiance. Ainsi, plus la confiance est élevée dans un mauvais résultat, plus le score augmente, signe d'une mauvaise interprétation.

Il est important de remarquer que l'indice multiplicateur est toujours plus grand dans le cas d'une mauvaise reconnaissance que dans le cas d'une bonne. Ainsi, quelle que soit la confiance accordée à un résultat, une mauvaise reconnaissance sera toujours plus pénalisée qu'une bonne reconnaissance.

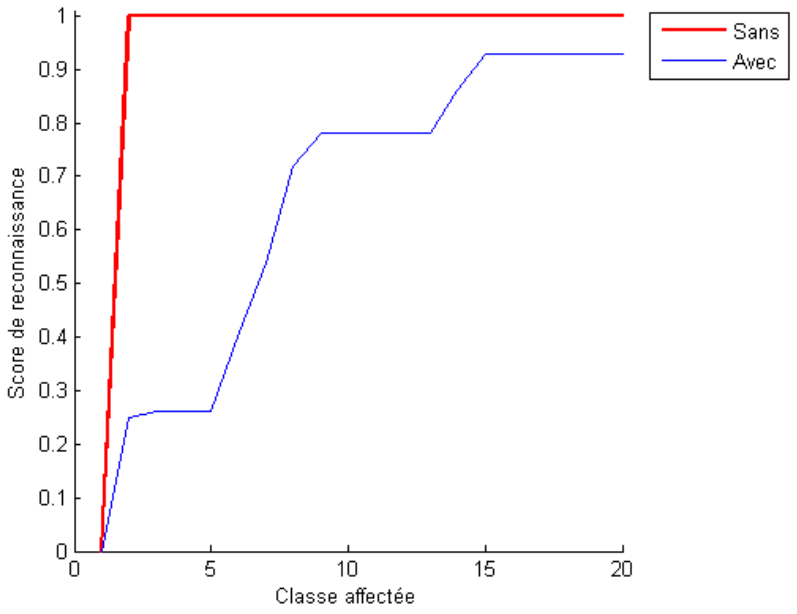
Pondération de la reconnaissance

Nous nous sommes enfin intéressés à l'effet d'une matrice des distances sur les résultats de reconnaissance. Pour cela, nous avons calculé la matrice de distance à partir de la taxonomie créée à la figure 4.10. La figure 4.23 nous montre l'évolution du score de reconnaissance en fonction de l'utilisation ou non d'une matrice des distances. La figure 4.23 (a) présente l'évolution du score de reconnaissance pour la classe 10 « plante en pot » en fonction de la classe assignée. On peut voir que l'utilisation d'une matrice de distance permet d'avoir un score plus fin que sans son utilisation. La figure 4.23 (b) présente l'évolution du score de reconnaissance en moyenne, pour les 20 classes, en fonction de la classe assignée. Nous avons trié les résultats afin de présenter l'évolution du score en fonction de la classe affectée de la

plus proche à la plus lointaine. Nous pouvons voir que cela confirme le fait que le score est mieux évalué avec une matrice de distance que sans.



(a) Résultat pour la classe 10 « plante en pot »



(b) Résultat en moyenne

FIGURE 4.23 – Valeur du score de reconnaissance avec et sans matrice de distance

4.3.3 Discussions

Les résultats que nous avons obtenus avec la méthode proposée sont satisfaisants. Nous avons vu que la métrique permet de bien prendre en compte : (i) une mauvaise localisation, (ii) une mauvaise reconnaissance et (iii) une mauvaise détection. Il est à noter que l'on pénalise les altérations dans l'ordre d'importance : d'abord la mauvaise détection, ensuite la reconnaissance puis la localisation. De même, dans les altérations possibles de localisation, on pénalise en priorité les problèmes de rotation et de translation, suivis des problèmes de mises à l'échelle et enfin des problèmes de perspective.

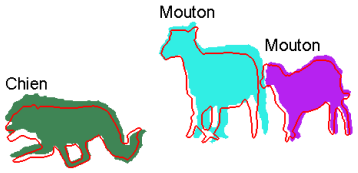
Nous avons également vu que la méthode est paramétrable et que cela influe sur les résultats d'évaluation. Le mode de mise en correspondance permet notamment de rendre la méthode d'évaluation plus sévère en augmentant le seuil. L'ajout de l'indice de confiance renvoyé par l'algorithme ainsi que l'utilisation de matrice des distances permet d'avoir un résultat d'évaluation plus fin.

Si nous reprenons les résultats d'interprétation de la figure 4.1, et que nous les évaluons, nous obtenons les résultats d'évaluation présentés à la figure 4.24. Nous pouvons voir que le résultat 3 est le moins bon, puisqu'il manque un objet. Suit le résultat 4, pour lequel il manque également un objet. Cependant, il s'agit d'un objet qui chevauche deux objets de la vérité terrain, ce qui permet d'avoir une évaluation moins pénalisante que pour le résultat 3. Les résultats 1 et 2 obtiennent des meilleurs scores d'évaluation. Le premier est meilleur que le second car il n'y a que des erreurs de localisation et aucune erreur de reconnaissance.

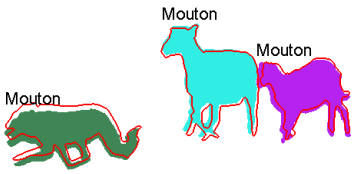
4.4 Conclusions

Nous avons travaillé sur la création d'une métrique permettant l'évaluation d'un résultat d'interprétation. Cette métrique permet de prendre en compte à la fois les informations concernant la localisation et la reconnaissance des objets dans la scène. Cette métrique se base sur une mise en correspondance, le calcul d'un score local à chaque objet mis en correspondance, puis le calcul d'un score global prenant en compte la sous et sur-détection.

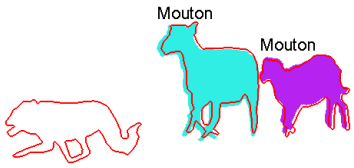
Nous avons créé cette métrique afin qu'elle soit paramétrable pour s'adapter à une application visée. Nous avons vu que la méthode de mise en correspondance peut être



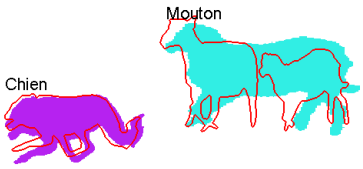
(a) Résultat 1 - score = 0,1242



(b) Résultat 2 - score = 0,1574



(c) Résultat 3 - score = 0,3716



(d) Résultat 4 - score = 0,1935

FIGURE 4.24 – Exemples d'évaluation de résultats d'interprétation sur une scène (Image originale tirée de [22]) : le contour rouge présente la vérité terrain de la localisation. Dans ce cas, le résultat 1 est jugé comme le meilleur par la méthode proposée

modifiée. De même, l'importance de la reconnaissance par rapport à la localisation peut être modifiée. Enfin, l'utilisation de matrice de distance, créée automatiquement ou manuellement, permet de grandement améliorer les performances de la méthode d'évaluation.

Les résultats obtenus par notre méthode d'évaluation correspondent aux objectifs que nous nous étions fixés. Nous avons vu qu'elle pénalise correctement les différentes altérations possibles.

Conclusions et perspectives

Bilan

Au cours de cette thèse, nous nous sommes intéressés à l'évaluation de résultats d'algorithmes d'interprétation d'images. Nous avons cherché à formaliser les algorithmes d'interprétation parmi lesquels les algorithmes de localisation et de reconnaissance jouent un rôle prépondérant Nous avons alors considéré l'interprétation d'images comme la juxtaposition d'un algorithme de localisation et d'un algorithme de reconnaissance d'objets, ce qui a conduit aux deux axes de recherche que nous avons développés.

Le premier axe de recherche sur lequel nous avons travaillé concerne l'évaluation d'un résultat de localisation ou de reconnaissance. Concernant l'évaluation de la localisation, nous avons proposé une méthodologie, se basant sur un ensemble de propriétés attendues, permettant de quantifier la performance des métriques d'évaluation des résultats de localisation. Ces propriétés, dont les 3 usuelles pour une distance ainsi que les 4 propriétés que nous avons définies, ont été proposées. La propriété de monotonie permet de s'assurer que plus un résultat est altéré, plus il est pénalisé. De même, la propriété de régularité permet de s'assurer que deux résultats proches vont être pénalisés de manière proche également. Nous avons appliqué cette méthodologie à la comparaison de *38* métriques en provenance de l'état de l'art sur une importante base de données synthétiques avec un total de *118 080* images altérées. Nous avons mis en évidence que les métriques provenant de la compétition Robin ne pouvaient pas être utilisées indépendamment les unes des autres. En cas de localisation par des boîtes englobantes, il est donc nécessaire d'utiliser et de combiner les résultats en provenance de trois métriques du projet Robin. Il est également ressorti le fait que les métriques topologiques permettent d'avoir un meilleur résultat que les métriques statistiques pour les métriques basées contour. Enfin, les métriques basées masque per-

mettent d'avoir une très bonne évaluation d'un résultat de localisation. En particulier, les deux métriques de Martin *et coll.* [47] ainsi que la métrique utilisée dans la compétition Pascal [22] ont montré leur intérêt. Ces travaux ont été publiés dans [76, 77, 78].

Nous avons également travaillé sur l'évaluation de la reconnaissance. Rappelons que la tâche a consisté à quantifier l'importance de l'erreur commise lorsqu'un objet est mal reconnu. La principale difficulté ici est que les classes sont identifiées par des variables qualitatives. Nous avons alors travaillé dans l'objectif de créer une matrice des distances entre classes d'objets. Pour cela, nous nous sommes intéressés aux outils de la reconnaissance de formes. Nous avons comparé différents modèles de représentation, plusieurs méthodes de calcul de similarité d'objets à partir de descripteurs. Il en ressort que le modèle des ensembles de points et la double association permettent d'avoir un bon taux de reconnaissance. En effet, dans le contexte difficile de la biométrie et avec une seule image en apprentissage, nous avons obtenu un taux d'erreur de l'ordre de 16% avec les méthodes testées. Nous avons ensuite comparé 8 descripteurs de points d'intérêt dans le cadre de la catégorisation. Nous avons pu voir que les descripteurs SIFT et ACP-SIFT permettent d'avoir des résultats satisfaisants tout en ayant un temps de calcul raisonnable. Ces travaux ont été publiés dans [79].

Le second axe de recherche a concerné le développement d'une méthode permettant d'évaluer globalement un résultat d'interprétation d'une image. La méthode proposée prend en compte, en même temps, la qualité de la localisation, de la reconnaissance et de la détection. Cette méthode est composée de quatre phases :

- une mise en correspondance,
- une évaluation locale pour chaque objet mis en correspondance,
- une prise en compte de la sous et de la sur-détection,
- un calcul du score global.

Nous avons validé cette méthode avec la base de données Pascal 2008 [22]. Cette base de données comporte *2 134* objets repartis dans 20 classes différentes. Nous avons vu que les performances sont satisfaisantes avec les paramètres par défaut, et permettent de pénaliser, du moins critique au plus critique, (i) la mauvaise localisation, (ii) la mauvaise reconnaissance et (iii) la mauvaise détection. Nous avons vu que cette méthode est paramétrable, notamment au niveau de la mise en correspondance et de l'évaluation locale. Ainsi, le seuil de la mise en correspondance permet d'avoir une méthode d'évaluation plus ou moins sévère. De même, l'importance de la reconnaissance par rapport à la localisation peut être adaptée à une application par-

ticulière. Les résultats obtenus par cette méthode d'évaluation globale sont corrects et répondent à nos attentes. Nous avons publié cette méthode dans [80].

Nous pouvons donc maintenant évaluer les résultats d'interprétation que nous présentions en introduction. La figure 4.25 présente les scores obtenus avec notre méthode d'évaluation et les paramètres par défaut. Ainsi, le résultat 3 est fortement pénalisé de part l'absence de détection de plantes en pot. Le résultat 4 obtient également un mauvais score en sur-détectant un objet. Les résultats 1 et 2 obtiennent de meilleurs scores. Le résultat 2 est pénalisé par la détection d'une seul plante à la place des deux présentes dans la vérité terrain, tandis que le résultat 1 est pénalisé par la mauvaise reconnaissance du chat.

Contributions

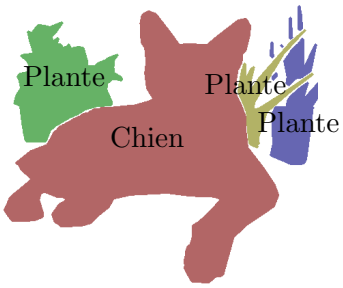
Nos contributions ont permis :
- de proposer une formalisation du concept de l'interprétation d'images,
- de définir un ensemble de propriétés que doit nécessairement satisfaire une métrique de localisation,
- de définir une méthodologie de calcul d'une distance entre des classes d'objets,
- de développer une méthode globale d'évaluation d'un résultat d'interprétation d'une image.

Perspectives

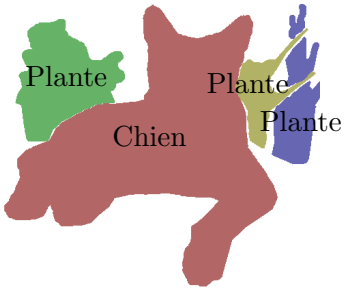
Les perspectives de cette thèse sont encore nombreuses. De nouvelles métriques de localisation peuvent être testées, le calcul des distances entre classes peut être amélioré.

Une des perspectives principale consiste en une évaluation subjective de la méthode d'évaluation globale. Pour cela, nous avons prévu de faire évaluer par des experts des résultats d'interprétation d'images. La comparaison des résultats d'interprétation provenant des experts avec ceux obtenus par notre méthode d'évaluation nous permettra de l'améliorer. Nous souhaitons notamment vérifier la pondération par défaut de la reconnaissance par rapport à la localisation et la valeur de la pénalisation en cas de sous et sur-détection. Une telle étude a été réalisée dans la thèse de Sébastien Chabrier [33] afin d'évaluer des résultats de segmentation. Un taux de bonne comparaison entre les experts et les métriques y est défini et un algorithme génétique permet alors d'optimiser les paramètres afin que l'évaluation soit en accord

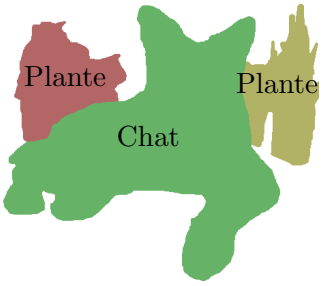
(a) Image originale



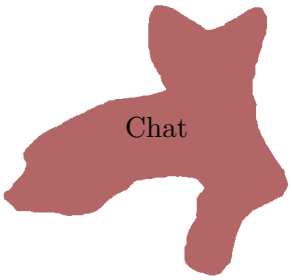
(b) Vérité terrain



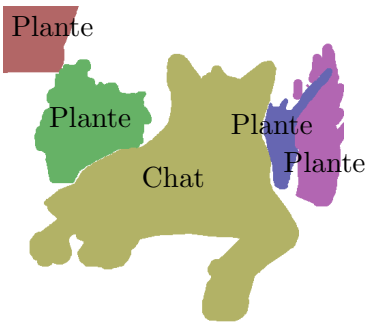
(c) Résultat 1 - score = 0,1016



(d) Résultat 2 - score = 0,0758



(e) Résultat 3 - score = 0,7694



(f) Résultat 4 - score = 0,254

FIGURE 4.25 – Évaluation de résultats d'interprétation d'une image. Dans ce cas, le résultat 2 est jugé comme le meilleur des quatre

avec l'attente des experts.

Enfin, une autre perspective est de développer une métrique d'évaluation globale qui serait non supervisée. En effet, nous avons vu que notre métrique d'évaluation nécessite obligatoirement une vérité terrain pour évaluer les résultats d'interprétation. Si cela est acceptable lors de compétition ou bien pour le développement d'un algo-

rithme, cela reste néanmoins une limitation. La question est donc de savoir s'il est possible de quantifier la qualité d'un résultat d'interprétation d'une image sans vérité terrain. Concernant la reconnaissance, la base de connaissance serait connue ce qui permettrait de facilement adapter notre méthode d'évaluation de la reconnaissance mise en place. Le calcul de similarité pourrait alors être effectué directement entre l'objet détecté et le modèle de classe à laquelle celui-ci est affecté. Par contre, il faudrait définir des métriques de bonne localisation des objets, en mode non supervisé, et être capable d'identifier une sur et sous-détection d'objets.

Publications de l'auteur

Chapitre de livre international

1. F. Cherifi, **B. Hemery**, R. Giot, M. Pasquet et C. Rosenberger, « Performance Evaluation Of Behavioral Biometric Systems », Book on Behavioral Biometrics for Human Identification : Intelligent Applications, 21 pages, 2009.

Conférences internationales avec comité de lecture et avec actes

1. **B. Hemery**, H. Laurent et C. Rosenberger, « Evaluation Metric for Image Understanding », *IEEE International Conference on Image Processing* (ICIP) 2009,
2. **B. Hemery**, H. Laurent, B. Emile et C. Rosenberger, « Comparative Study Of Local Descriptors For Measuring Object Taxonomy », *International Conference on Image and Graphics* (ICIG) 2009,
3. **B. Hemery**, H.Laurent, C. Rosenberger et B. Emile, « Evaluation Protocol for Localization Metrics - Application to a Comparative Study », *International Conference on Image and Signal Processing* (ICISP), 2008,
4. **B. Hemery**, J. Mahier, M. Pasquet et C. Rosenberger, « Face Authentication for Banking », *First International Conference on Advances in Computer-Human Interaction* (ACHI), 2008,
5. **B. Hemery**, H. Laurent et C. Rosenberger, « Comparative study of evaluation metrics of object localization by bounding boxes », *International conference on Image and Graphics* (ICIG), pages 459 – 464, 2007,
6. **B. Hemery**, C. Rosenberger et H. Laurent, « The ENSIB database : a benchmark for face recognition », *International Symposium on Signal Processing*

and its Applications (ISSPA), special session « Performance Evaluation and Benchmarking of Image and Video Processing », 2007,

7. **B. Hemery**, C. Rosenberger, C. Toinard et B. Emile, « Comparative study of invariant descriptors for face recognition », 8th *International IEEE Conference on Signal Processing* (ICSP), 2006.

Conférences nationales avec comité de lecture et avec actes

1. R. Belguechi,**B. Hemery**, et C. Rosenberger. « Authentification révocable pour la vérification basée texture d'empreintes digitales ». Actes du 17eme congrés de Reconnaissance des Formes et Intelligence Artificielle (RFIA) 2010,
2. **B. Hemery**, H. Laurent et C. Rosenberger, « Étude comparative de métriques pour l'évaluation de la localisation d'objets par des boîtes englobantes », *colloque GRETSI* 2007,
3. **B. Hemery**, H. Laurent et C. Rosenberger, « Reconnaissance faciale par des invariants locaux », *Congrès des jeunes chercheurs en vision par ordinateur* (ORASIS) 2007.

Conférences nationales sans comité de lecture

1. **B. Hemery**, H. Laurent et C. Rosenberger, « Critère d'évaluation globale d'un résultat d'interprétation d'images », GDR-ISIS groupe SCATI, réunion du 2 avril 2009,
2. **B. Hemery**, H. Laurent et C. Rosenberger, « Protocole d'étude de métriques d'évaluation de la localisation », GDR-ISIS, réunion du 17 décembre 2008,
3. **B. Hemery**, « Évaluation d'algorithmes d'interprétation d'images », Séminaire d'équipe ISS de l'Institut PRISME, 12 juin 2008.

Rapports de recherche

1. **B. Hemery** : « Rapport de fin de 1ère année de thèse », *Rapport de recherche, décembre 2007*,
2. **B. Hemery** : « Étude bibliographique sur l'évaluation de l'interprétation d'images », *Rapport de recherche, décembre 2006*,

3. **B. Hemery** : « Reconnaissance faciale pour l'authentification biométrique », *Rapport de stage de master recherche, août 2006.*

Article soumis

1. **B. Hemery**, H. Laurent, B. Emile, C. Rosenberger, « Comparative Study of Localization Metrics for the Evaluation of Image Interpretation Systems », *Journal of Electronic Image.*

Bibliographie

[1] Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, http ://authors.library.caltech.edu/7694, 2007. [cité p. 2, 108, 109, 110, 111, 128, 170]

[2] E. D'Angelo, S. Herbin, and M. Ratiéville. Robin challenge evaluation principles and metrics, 11 2006. http ://robin.inrialpes.fr. [cité p. 7, 40, 58, 64, 79, 117, 162]

[3] Paul Viola and Michael J. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2) :137–154, 2002. [cité p. 9]

[4] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis And Machine Intelligence (PAMI)*, 27 :1615–1630, 2005. [cité p. 9, 37, 108]

[5] Frédéric Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision (ICCV)*, volume 1, 2005. [cité p. 9, 12, 38]

[6] Herbert Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. *European Conference on Computer Vision*, 9, 5 2006. [cité p. 9]

[7] Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms : a systematic survey. *IEEE Transactions on Image Processing*, 14(3) :294–307, 2005. [cité p. 9]

[8] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(10) :1337–1342, 2003. [cité p. 9]

[9] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, and Hélène Laurent. Review and evaluation of commonly-implemented background subtraction algorithms. In *International Conference on Pattern Recognition (ICPR)*, 2008. [cité p. 10]

[10] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Improving a discriminative approach to object recognition using image patches. *Lecture Notes In Computer Science*, 3663 :326–333, 2005. [cité p. 12, 97]

[11] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories : An in-depth study. Technical report, INRIA, 2005. [cité p. 12]

[12] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision (ECCV)*, pages 1–22, 2004. [cité p. 12, 13, 38, 167]

[13] Christophe Garcia and Manolis Delakis. Convolutional face finder : a neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(11) :1408–1423, 2004. [cité p. 14, 15]

[14] Christophe Garcia and Manolis Delakis. A neural architecture for fast and robust face detection. *IEEE-IAPR International Conference on Pattern Recognition (ICPR)*, 16 :40–43, 2002. [cité p. 14, 15]

[15] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1 :886–893, 2005. [cité p. 14, 15]

[16] Frédéric Jurie and Cordelia Schmid. Scale-invariant shape features for recognition of object categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2 :90–96, 2004. [cité p. 15, 16]

[17] Mario Fritz, Bastian Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2005. [cité p. 16]

[18] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision (ECCV)*, pages 17–32, 2004. [cité p. 16]

[19] G. Dorko and C. Schmid. Object class recognition using discriminative local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(1) :1–13, 2004. [cité p. 16]

[20] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features : efficient boosting procedures for multiclass object detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2 :762–769, 2004. [cité p. 16, 17]

[21] Christophe Rosenberger. Contribution à l'évaluation d'algorithmes de traitement d'images. Habilitation à diriger la recherche de l'université d'Orléans, 12 2006. [cité p. 22]

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http ://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html. [cité p. 32, 116, 117, 119, 120, 127, 140, 142]

[23] Pascaline Parisot. *Suivi d'objets dans des séquences d'images de scènes déformables.* PhD thesis, Université de Toulouse, 2009. [cité p. 32]

[24] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50, 1988. [cité p. 33, 97]

[25] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004. [cité p. 34, 36, 97, 108]

[26] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or" how do i organize my holiday snaps ?". *Lecture Notes In Computer Science*, 2350 :414–431, 2002. [cité p. 35, 108, 168]

[27] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(9) :891–906, 1991. [cité p. 35, 36, 108, 168]

[28] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6) :367–375, 1987. [cité p. 35, 108]

[29] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157. Kerkyra, Greece, 1999. [cité p. 36]

[30] Yan Ke and Rahul Sukthankar. Pca-sift : A more distinctive representation for local image descriptors. In *IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR)*, volume 2. IEEE Computer Society ; 1999, 2004. [cité p. 37, 108]

[31] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context : A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000. [cité p. 37, 108]

[32] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6) :679–698, 1986. [cité p. 37]

[33] Sébastien Chabrier. *Contribution à l'évaluation de performances en segmentation d'images.* PhD thesis, Université d'Orléans, 12 2005. [cité p. 40, 41, 42, 43, 52, 58, 79, 143, 162, 163]

[34] A. J. Baddeley. An error metric for binary images. In *International Workshop on Robust Computer Vision*, pages 59–78, 3 1992. [cité p. 41, 44, 79, 162, 163]

[35] D. Coquin, P. Bolon, and Y. Chehadeh. Evaluation quantitative d'images filtrées. *GRETSI*, 2 :1351–1354, 1997. [cité p. 42, 79, 162]

[36] Dale L. Wilson, Adrian J. Baddeley, and Robyn A. Owens. A new metric for grey-scale image comparison. *International Journal of Computer Vision*, 24(1) :5–17, 1997. [cité p. 42, 44, 79, 162, 163]

[37] Michèle Basseville. Distance measures for signal processing and pattern recognition. *Signal Processing*, 18(4) :349–369, 1989. [cité p. 42, 79, 162]

[38] Olivier Michel, Richard G. Baraniuk, and Patrick Flandrin. Time-frequency based distance and divergence measures. In *IEEE International Symposium on TF and TS analysis*, pages 64–67, 1994. [cité p. 42]

[39] T. Peli and D. Malah. A study of edge detection algorithms. In *Computer Graphics and Image Processing*, 1982. [cité p. 43, 79, 162]

[40] W. Pratt, O. D. Faugeras, and A. Gagalowicz. Visual discrimination of stochastic texture fields. *IEEE Transactions on Systems, Man, and Cybernetics (SMC)*, 8(11) :796–804, 1978. [cité p. 43, 79, 162]

[41] M. Beauchemin, KP. B. Thomson, and G. Edwards. On the hausdorff distance used for the evaluation of segmentation results. *Canadian Journal of Remote Sensing*, 24(1) :3–8, 1998. [cité p. 43, 79, 162]

[42] Christophe Odet, Boubakeur Belaroussi, and Hugues Benoit-Cattin. Scalable discrepancy measures for segmentation evaluation. *IEEE International Conference on Image Processing (ICIP)*, 1 :785–788, 9 2002. [cité p. 44, 79, 163]

[43] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M.Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G.Dorko, et al. The 2005 pascal visual object classes challenge, 2005. http ://www.pascal-network.org/challenges/VOC/. [cité p. 45, 60, 62, 63, 79, 163]

[44] Olof Henricsson and Emmanuel Baltsavias. 3-d building reconstruction with aruba : A qualitative and quantitative evaluation. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 65–76. Verlag, 1997. [cité p. 46, 79, 163]

[45] W. A. Yasnoff, J. K. Mui, and J. W. Bacus. Error measures for scene segmentation. *Pattern Recognition*, 9 :217–231, 1977. [cité p. 46, 79, 163]

[46] Christian Wolf and Jean-Michel Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8(4) :280–296, 2006. [cité p. 47, 58, 79, 121, 163]

[47] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms

and measuring ecological statistics. *International Conference on Computer Vision (ICCV)*, 2 :416–423, 7 2001. [cité p. 49, 50, 79, 95, 116, 121, 142, 163]

[48] Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. *IEEE International Conference on Image Processing (ICIP)*, 3 :53–56, 1995. [cité p. 50, 79, 163]

[49] Adel Hafiane. *Caractérisation de textures et segmentation pour la recherche d'images par le contenu.* PhD thesis, Université de Paris-Sud XI, 12 2005. [cité p. 50, 79, 164]

[50] Adel Hafiane, Sébastien Chabrier, Christophe Rosenberger, and Hélène Laurent. A new supervised evaluation criterion for region based segmentation methods. In *International conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 439–448, 2007. [cité p. 51, 79, 164]

[51] Laurent Vinet. *Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques.* PhD thesis, Université de Paris IX Dauphine, 7 1991. [cité p. 51, 79, 164]

[52] J.-P. Cocquerez and S. Philipp. *Analyse d'Images : filtrage et segmentation.* Masson, 1995. [cité p. 51, 79, 164]

[53] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5 :32, 1957. [cité p. 55, 99, 121]

[54] K. Riesen, M. Neuhaus, and H. Bunke. Bipartite graph matching for computing the edit distance of graphs. *Lecture Notes in Computer Science : Graph-Based Representations in Pattern Recognition*, 4538 :1–12, 2007. [cité p. 55, 99]

[55] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4) :255–259, 1998. [cité p. 56]

[56] M.L. Fernández and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 22(6-7) :753–758, 2001. [cité p. 56]

[57] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12) :2270–2285, 2005. [cité p. 57]

[58] Raffaele Cappelli, Dario Maio, and Davide Maltoni. Combining fingerprint classifiers. In *International Workshop on Multiple Classifier Systems*, pages 351–361, London, UK, 2000. Springer-Verlag. [cité p. 57]

[59] Silogic Inria. Etiseo metrics definition, 01 2006. [cité p. 58]

[60] Warren Sturgis McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–133, 1943. [cité p. 59]

[61] Bernhard Schölkopf and Alexander Smola. Learning with kernels, 2007. [cité p. 59]

[62] Stefan Wender and Klaus C. J. Dietmayer. An adaptable object classification framework. In *IEEE Intelligent Vehicules Symposium*, pages 150–155, 2006. [cité p. 60]

[63] James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1) :29–36, 1982. [cité p. 60]

[64] BioAPI Consortium. Information technology – biometric performance testing and reporting – part 1 : Principles and framework. Technical report, ISO/IEC 19795-1, 2006. [cité p. 60, 61, 62]

[65] Henning Muller, Wolfgang Muller, David McG. Squire, and Thierry Pun. Performance evaluation in content-based image retrieval : Overview and proposals. *Pattern Recognition Letters*, 22(5) :593–601, 2001. [cité p. 62, 63]

[66] Michael C. Steele. *The Power of Categorical Goodness-Of-Fit Statistics*. PhD thesis, Griffith University, 2003. [cité p. 65, 67, 68]

[67] Armelle Brun and Kamel Smaïli. Fiabilité de la référence humaine dans la détection de thème. In *Traitement Automatique des Langues Naturelles*, 2004. [cité p. 68]

[68] Jean Carletta. Squibs and discussions, assessing agreement on classification tasks : The kappa statistic. *Computational Linguistics*, 22(2) :249–254, 1996. [cité p. 68]

[69] D.K. Krippendorff. *Content Analysis : An Introduction to Its Methodology*. Sage Pubns, 2004. [cité p. 69]

[70] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000. [cité p. 97]

[71] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1) :63–86, 2004. [cité p. 97, 107]

[72] Christophe Rosenberger and Luc Brun. Similarity-based matching for face authentication. In *International Conference on Pattern Recognition (ICPR)*, 2008. [cité p. 97]

[73] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17, 2007. [cité p. 98]

[74] Aleix M. Martinez and Robert Benavente. The ar face database. *CVC Technichal Report*, 24, 1998. [cité p. 100, 101]

[75] Ihsin T. Phillips and Atul K. Chhabra. Empirical performance evaluation of graphics recognition systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(9) :849–870, 1999. [cité p. 119, 121]

[76] Baptiste Hemery, Hélène Laurent, and Christophe Rosenberger. Comparative study of evaluation metrics of object localization by bounding boxes. In *Proceedings of*

the International Conference on Image and Graphics (ICIG), pages 459 – 464, 2007. [cité p. 142]

[77] Baptiste Hemery, Hélène Laurent, and Christophe Rosenberger. Étude comparative de métriques pour l'évaluation de la localisation d'objets par des boîtes englobantes. In *Colloque GRETSI*, pages 459 – 464, 2007. [cité p. 142]

[78] Baptiste Hemery, Hélène Laurent, Christophe Rosenberger, and Bruno Emile. Evaluation protocol for localization metrics - application to a comparative study. In Springer Berlin / Heidelberg, editor, *Proceedings of the 3rd International Conference on Image and Signal Processing (ICISP)*, volume 5099/2008, pages 273–280, 2008. [cité p. 142]

[79] Baptiste Hemery, Hélène Laurent, Bruno Emile, and Christophe Rosenberger. Comparative study of local descriptors for measuring object taxonomy. In *Proceedings of the International Conference on Image and Graphics (ICIG)*, 2009. [cité p. 142]

[80] Baptiste Hemery, Hélène Laurent, and Christophe Rosenberger. Evaluation metric for image understanding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2009. [cité p. 143]

Annexe

Annexe A

Métriques de localisation

TABLE A.1: Liste des métriques de localisation

Nom	Métrique	Représentation	Formule	Paramètres	Référence
Robin localisation	ROB_{loc}	Box	$ROB_{loc}(BB_{vt}, BB_l) = \frac{2}{\pi} arctan(max(\frac{\lvert x_l - x_{vt}\rvert}{w_{vt}}, \frac{\lvert y_l - y_{vt}\rvert}{h_{vt}}))$		[2]
Robin complétude	ROB_{com}	Box	$ROB_{com}(BB_{vt}, BB_l) = \frac{\lvert \mathcal{A}_l - \mathcal{A}_{vt}\rvert}{max(\mathcal{A}_l, \mathcal{A}_{vt})}$		[2]
Robin correction	ROB_{cor}	Box	$ROB_{cor}(BB_{vt}, BB_l) = \frac{2}{\pi} arctan(\lvert \frac{h_l}{w_l} - \frac{h_{vt}}{w_{vt}}\rvert)$		[2]
	$ErrLoc$	Frontière	$ErrLoc(I_{vt}, I_l) = \frac{Card((I^{Fr}_{vt\setminus l}) \cup (I^{Fr}_{l\setminus vt}))}{Card(I)}$		[34, 33]
	$ErrSous$	Frontière	$ErrSous(I_{vt}, I_l) = \frac{Card(I^{Fr}_{vt\setminus l})}{Card(I^{Fr}_{vt})}$		[34, 33]
	$ErrSur$	Frontière	$ErrSur(I_{vt}, I_l) = \frac{Card(I^{Fr}_{l\setminus vt})}{Card(I) - Card(I^{Fr}_{vt})}$		[34, 33]
Rapport Signal Bruit	SNR	Frontière	$SNR(I_{vt}, I_l) = \left[\frac{1}{Card(I)} \sum_{k \in I} \frac{g_{I_l}(k)^2}{(g_{I_{vt}}(k) - g_{I_l}(k))^2} \right]^{\frac{1}{2}}$		[36, 35]
Root Mean Square	RMS	Frontière	$RMS(I_{vt}, I_l) = \left[\frac{1}{Card(I)} \sum_{k \in I} (g_{I_{vt}}(k) - g_{I_l}(k))^2 \right]^{\frac{1}{2}}$		[36, 35]
Distance Lq	Lq	Frontière	$L_q(I_{vt}, I_l) = \left[\frac{1}{Card(I)} \sum_{k \in I} \lvert g_{I_l}(k) - g_{I_{vt}}(k)\rvert^q \right]^{\frac{1}{q}}$	$q \in \{1, 3\}$	[36, 35]
Distance de Kullback	KUL	Frontière	$KUL(I_{vt}, I_l) = \frac{1}{Card(I)} \sum_{k \in I} \left(g_{I_l}(k) - g_{I_{vt}}(k)\right) * Log\left(\frac{g_{I_l}(k)}{g_{I_{vt}}(k)}\right)$		[33, 37]
Distance de Bhattacharyya	BAH	Frontière	$BHA(I_{vt}, I_l) = -Log\left(\frac{1}{Card(I)} \sum_{k \in I} \sqrt{g_{I_{vt}}(k) * g_{I_l}(k)}\right)$		[33, 37]
Distance de Jensen	JEN	Frontière	$JEN(I_{vt}, I_l) = J\left(\frac{I_{vt} + I_l}{2}, I_{vt}\right)$; $J(I_1, I_2) = H_\alpha(\sqrt{I_1 * I_2}) - \frac{H_\alpha(I_1) + H_\alpha(I_2)}{2}$ $H_\alpha(I_1) = \frac{1}{1-\alpha} Log_2\left(\sum_{k \in I_1} (g_{I_1}(k))^\alpha\right)$	$\alpha = 3$	[33, 37]
Distance Moyenne	$DMoy$	Frontière	$DMoy(I_{vt}, I_l) = \frac{1}{Card(I^{Fr}_l)} \sum_{k \in I^c_l} d(k, I^c_{vt})$		[39]
Carré de la distance moyenne	$DMoC$	Frontière	$DMoC(I_{vt}, I_l) = \frac{1}{Card(I^{Fr}_l)} \sum_{k \in I^{Fr}_l} d(k, I^c_{vt})^2$		[39]
Figure of Merit	FOM	Frontière	$FOM(I_{vt}, I_l) = \frac{1}{MP} \sum_{k \in I^F r_l} \frac{1}{1 + \alpha * d(k, I^F r_{vt})^2}$	$\alpha = \frac{1}{9}$	[40, 39]
Distance de Hausdorff	HAU	Frontière	$HAU(I_{vt}, I_l) = \max\left(h(I_{vt}, I_l) \ , \ h(I_l, I_{vt}) \right)$		[34, 41]

TABLE A.1: Liste des métriques de localisation (Suite)

Nom	Métrique	Représentation	Formule	Paramètres	Référence
			$h(I_1, I_2) = \max_{k_1 \in I_1^c} \min_{k_2 \in I_2^c} d(k_1, k_2)$		
Distance de Baddeley	BAD	Frontière	$BAD(I_{vt}, I_l) = \left[\frac{1}{card(I)} \sum_{k \in I_{vt}^{Fr} \cup I_l^c} \lvert d(k, I_{vt}^{Fr}) - d(k, I_l^{Fr})\rvert^P\right]^{\frac{1}{P}}$ with $P \geq 1$	$P \in \{1, 2, 3\}$	[34, 36]
Odet	ODI_n	Frontière	$ODI_n(I_{vt}, I_l) = \frac{1}{Card(I_{l\setminus vt}^{Fr})} \sum_{k \in I_{l\setminus vt}^{Fr}} \left(\frac{d(k, I_{vt}^{Fr})}{d_{Th}}\right)^n$	$n \in \{1, 2\}$; $d_{Th} = 5$	[33, 42]
Odet	UDI_n	Frontière	$UDI_n(I_{vt}, I_l) = \frac{1}{Card(I_{vt\setminus l}^{Fr})} \sum_{k \in I_{vt\setminus l}^{Fr}} \left(\frac{d(k, I_l^{Fr})}{d_{Th}}\right)^n$	$n \in \{1, 2\}$; $d_{Th} = 5$	[33, 42]
Pascal	PAS	Masque	$PAS(I_{vt}, I_l) = \frac{Card(I_{vt}^{Re} \cap I_l^{Re})}{Card(I_{vt}^{Re} \cup I_l^{Re})}$		[43]
Henricsson	$HEN1$	Masque	$HEN1(I_{vt}, I_l) = \frac{\lvert Card(I_l^{Re}) - Card(I_{vt}^{Re})\rvert}{Card(I_{vt}^{Re})}$		[44]
Henricsson	$HEN2$	Masque	$HEN2(I_{vt}, I_l) = \frac{Card(I_{l\setminus vt}^{Re}) + Card(I_{vt\setminus l}^{Re})}{Card(I_{vt}^{Re})}$		[44]
Yasnoff	$YAS1$	Masque	$YAS_1(I_{vt}, I_l, k) = 100 * \frac{Card(I_l^{Re(k)}) - Card(I_{vt\cap l}^{Re(k)})}{Card(I_l^{Re(k)})}$		[33, 45]
Yasnoff	$YAS2$	Masque	$YAS_2(I_{vt}, I_l, k) = 100 * \frac{Card(I_{vt}^{Re(k)}) - Card(I_{vt\cap l}^{Re(k)})}{Card(I) - Card(I_l^{Re(k)})}$		[33, 45]
Yasnoff	$YAS3$	Masque	$YAS_3(I_{vt}, I_l, k) = \frac{100}{Card(I)} \sqrt{\sum_{a \in I_{l\setminus vt}^{Re(k)}} d(a, I_{vt}^{Re(k)})}$		[33, 45]
Précision niveau pixel	P_{px}	Masque	$Ppx(I_{vt}, I_l) = \frac{Card(I_{vt}^{Re} \cap I_l^{Re})}{Card(I_l^{Re})}$		[46]
Rappel niveau pixel	R_{px}	Masque	$Rpx(I_{vt}, I_l) = \frac{Card(I_{vt}^{Re} \cap I_l^{Re})}{Card(I_{vt}^{Re})}$		[46]
Martin - global consistency error	MAR_{gce}	Masque	$MAR_{gce}(I_{vt}, I_l) = \frac{1}{Card(I)} \min\left(\sum_{k \in I} E(I_{vt}, I_l, k), \sum_{k \in I} E(I_l, I_{vt}, k)\right)$ $E(I_1, I_2, k) = \frac{Card(I_{1\setminus 2}^{Re(k)})}{Card(I_1^{Re(k)})}$		[47, 33]
Martin - local consistency error	MAR_{lce}	Masque	$MAR_{lce}(I_{vt}, I_l) = \frac{1}{Card(I)} \sum_{k \in I} \min(E(I_{vt}, I_l, k), E(I_l, I_{vt}, k))$		[47, 33]
Distance de Hamming	HAM	Masque	$HAM(I_{vt}, I_l) = 1 - \frac{D_H(I_{vt}, I_l) + D_H(I_l, I_{vt})}{2 * Card(I)}$		[33, 48]

TABLE A.1: Liste des métriques de localisation (Suite)

Nom	Métrique	Représentation	Formule	Paramètres	Référence
Hafiane	$HAF1$	Masque	$HAF1(I_{vt}, I_l) = \eta \sum_{i,\text{argmax}_j\, Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})} \frac{Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})}{Card(I_{vt}^{Re(i)} \cup I_l^{Re(j)})}$ $\eta = \frac{N^*(I_{vt})}{N(I_l)}$ if $N(I_l) \geq N(I_{vt})$ ou $\eta = \frac{1}{Log\left(\frac{N(I_{vt})}{N(I_l)}\right)}$ sinon	$N(I) =$ nombre d'objet dans I	[49]
Hafiane	$HAF2$	Masque	$HAF2(I_{vt}, I_l) = \frac{M(I_{vt}, I_l) + m \times \eta}{1+m}$ $M(I_{vt}, I_l) = \sum_{j,\text{argmax}_i\, Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})} \frac{Card(I_{vt}^{Re(i)} \cap I_l^{Re(j)})}{Card(I_{vt}^{Re(i)} \cup I_l^{Re(j)})} \rho_j$	$m = 0,2$; $\rho_j = \frac{Card(I_l^{Re(j)})}{Card(I)}$	[50]
Vinet	VIN	Masque	$VIN(I_{vt}, I_l) = Card(I) - \sum_{C'} Card(I_l^{Re(i)} \cap I_{vt}^{Re(j)})$		[51, 52]

Liste des abréviations

B	Base de données d'images
$N^*(B)$	Nombre d'images dans une base de données
$N^*(I)$	Nombre d'objets dans l'image I
$\mathcal{A}(B)$	Annotation de la base de données
$\mathcal{Y}$	Espace des classes
$N^*(\mathcal{Y})$	Nombre de classes dans l'espace des classes
$\mathcal{Y}^+$	Espace des classes étendu aux classes *Autre* et *Ambigu*
$\mathcal{Z}$	Espace des localisations
$(Y_i^*, Z_i^*)_j$	Classe et localisation de l'objet i dans l'image j provenant de la vérité terrain
$(Y_i, Z_i)_j$	Classe et localisation de l'objet i dans l'image j provenant d'un résultat d'interprétation
λ	Seuil de la fonction de décision
μ_i	Confiance accordée à un résultat d'interprétation
I_l	Image contenant un résultat de localisation Z_i
I_{vt}	Image contenant la vérité terrain Z_i^*
$k = (x, y)$	Pixel k de coordonnées (x, y) dans une image
$g_I(k) = g_I(x, y)$	Niveau de gris du pixel k dans l'image I
I^{Fr}	Ensemble des pixels appartenant à la frontière dans l'image I
I^{Fr_i}	Ensemble des pixels appartenant à la frontière de l'objet i dans l'image I
$I^{Fr}_{1\setminus 2}$	Ensemble des pixels appartenant à la frontière de l'image I_1 mais pas à la frontière de l'image I_2
$I^{Fr}_{1\cap 2}$	Ensemble des pixels appartenant à la frontière de l'image I_1 et I_2
$I^{Fr}_{1\cup 2}$	Ensemble des pixels appartenant à la frontière de l'image I_1 ou I_2
I^{Re}	Ensemble des pixels de la région dans l'image I
$I^{Re(i)}$	Ensemble des pixels de la région de l'objet i dans l'image I

Table des figures

Liste des tableaux

Les algorithmes de traitement d'images regroupent un ensemble de méthodes qui vont traiter l'image depuis son acquisition par un capteur jusqu'à l'extraction de l'information utile pour une application donnée. Parmi ceux-ci, les algorithmes d'interprétation ont pour but de détecter, localiser et reconnaître un ou plusieurs objets dans une image. Le problème traité dans cette thèse réside dans l'évaluation de résultats d'interprétation d'une image ou une vidéo lorsque l'on dispose de la vérité terrain associée. Les enjeux sont nombreux comme la comparaison d'algorithmes, l'évaluation d'un algorithme au cours de son développement ou son paramétrage optimal.

Nous proposons dans cette thèse une formalisation des propriétés attendues d'une métrique de localisation. Nous réalisons une étude comparative rigoureuse des métriques de localisation de l'état de l'art au vu de ces propriétés. Nous réalisons un travail similaire sur les méthodes de reconnaissance utilisant une représentation locale des objets dans le but de quantifier une erreur de reconnaissance.

Nous avons mis au point une méthode d'évaluation d'un résultat d'interprétation d'une image exploitant les leçons de ces études comparatives. L'avantage de la méthode proposée est de pouvoir évaluer un résultat d'interprétation d'une image en prenant en compte à la fois la qualité de la localisation, de la reconnaissance et de la détection d'objets d'intérêt dans l'image. Le comportement de cette méthode d'évaluation a été testé sur une large base de tests et s'avère intéressant. Plusieurs paramètres permettent de modifier le comportement de cette méthode suivant l'application visée.

Evaluation of Image Interpretation

Image processing algorithms include a set of methods that process the image from its acquisition by a sensor to the extraction of useful information for a given application. Among these image interpretation algorithms, some are designed to detect, localize and identify one or more objects in an image. The problem addressed in this thesis is the evaluation of interpretation results of an image or a video, given the associated ground truth. Challenges are multiple such as the comparison of algorithms, evaluation of an algorithm during its development or its optimal setting.

We propose in this thesis a formalization of desired properties for a localization metric. We make a rigorous comparative study of localization metric of the state of the art in view of these properties. We carry out similar work on recognition methods using a local representation of objects in order to quantify a recognition error.

We have developed a method for evaluating an interpretation result of an image making use of the lessons from these studies. The advantage of the proposed method is to evaluate an image interpretation result by taking into account at the same time the quality of the localization, recognition and detection of objects of interest in the image. The behavior of this evaluation method has been tested on a database and is interesting. Several parameters allow us to change the behavior of this method for a given application.

Indexation Rameau : Evaluation / Reconnaissance De Formes (Informatique) / Traitement D'Images – Techniques Numériques / Classification
Indexation libre : Évaluation, Interprétation d'images, Localisation, Reconnaissance, Détection

Traitement du signal et des images

Laboratoire GREYC - UMR CNRS 6072 - Université de Caen Basse-Normandie - Ensicaen
6 Boulevard du Maréchal Juin - 14050 CAEN CEDEX