THÈSE

présentée pour l'obtention du grade de :

Docteur de TELECOM ParisTech

Spécialité : **Traitement du Signal et des Images**

par :

Rachid BENMOKHTAR

Ecole Doctorale d'Informatique, Télécommunications et Electronique de Paris (EDITE)

Titre de la thèse :

Fusion multi-niveaux pour l'indexation et la recherche multimédia par le contenu sémantique

Soutenue le 9 Juin 2009 devant le jury composé de :

Gaël	RICHARD	TELECOM ParisTech	Président du jury
Hervé Philippe	GLOTIN MULHEM	Université du Sud Toulon-Var LIG de Grenoble	Rapporteurs
Peter Bernard	SANDER MERIALDO	Polytechnique de Nice Eurécom	Examinateurs
Sid-Ahmed	BERRANI	Orange-France Télécom	Invité
Benoit	HUET	Eurécom	Directeur de thèse

Résumé

Aujourd'hui, l'accès aux documents dans les bases de données, d'archives et sur Internet s'effectue principalement grâce à des données textuelles : nom de l'image ou mots-clés. Cette recherche est non exempte de fautes plus ou moins graves : omission, orthographe, etc. Les progrès effectués dans le domaine de l'analyse d'images et de l'apprentissage automatique permettent d'apporter des solutions comme l'indexation et la recherche à base des caractéristiques telles que la couleur, la forme, la texture, le mouvement, le son et le texte. Ces caractéristiques sont riches en informations, notamment, d'un point de vue sémantique.

Cette thèse s'inscrit dans le cadre de l'indexation automatique par le contenu sémantique des documents multimédia : plans vidéos et images-clés. L'indexation consiste à extraire, représenter et organiser efficacement le contenu des documents d'une base de données.

Cependant, l'indexation est confrontée au «fossé sémantique» qui sépare les représentations visuelles brutes (bas-niveau) et conceptuelles (haut-niveau). Pour limiter les conséquences de cette problématique, nous avons introduit dans le système différents types de descripteurs, tout en prenant à notre avantage les avancées scientifiques dans le domaine de l'apprentissage automatique et de la "fusion multi-niveaux". En effet, la fusion est utilisée dans le but de combiner des informations hétérogènes issues de plusieurs sources afin d'obtenir une information globale, plus complète, de meilleure qualité, permettant de mieux décider et d'agir. Elle peut être appliquée sur plusieurs niveaux du processus de classification. Dans ce mémoire, nous avons étudié la fusion des caractéristiques "*feature fusion*", la fusion de classifieurs "*classifier fusion*" ainsi qu'à un niveau décisionnel dit de raisonnement "*decision fusion*".

Tout d'abord, nous présenterons les techniques permettant d'exploiter la fusion de classifieurs dans le système d'indexation et de recherche, en particulier par la théorie des évidences adaptée au réseau de neurones, donnant ainsi ce que nous appelons *Neural Network based on Evidence Theory (NNET)*. Cette théorie a l'avantage de présenter deux nouvelles informations importantes pour la prise de décision comparée aux méthodes probabilistes : l'ignorance du système et le degré de croyance. Ensuite, le NNET a été amélioré en intégrant les relations entre descripteurs et concepts, modélisées par un vecteur de pondération basé sur l'entropie et la perplexité. La combinaison de ce vecteur avec les sorties de classifieurs, nous donne un nouveau modèle, que nous appelons *Perplexity based Evidential Neural Network (PENN)*.

Par ailleurs, nous avons introduit le thème important des ontologies et de la similarité inter-concepts. C'est à dire l'étude des relations entre les classes. En général, les concepts ne

sont pas exprimés en général de manière isolée et une forte corrélation existe entre certaines classes. Une première difficulté réside dans l'utilisation d'une ontologie qui décrit les relations existantes entre les concepts. La deuxième difficulté qui nous intéresse plus, réside dans l'exploitation de cette information sémantique. Trois types d'informations ont été utilisés : les descripteurs visuels, la cooccurrence et la similarité sémantique, en conjonction avec une base de connaissance multimédia pour l'interprétation sémantique des plans vidéo. Le système final sera dénommé *Ontological PENN*.

Enfin, nous répondrons à la question qui concerne l'utilité de la fusion bas-niveau. Ceci n'a été possible qu'après une étude statistique des données avant et après la fusion des caractéristiques. Les systèmes proposés ont été validés sur les données de TRECVid (projet NoE K-Space) et les vidéos de football fournies par Orange-France Télécom Labs (projet CRE-Fusion).

Mots-clés : Indexation des plans vidéo, fossé sémantique, classification, fusion de descripteurs, fusion de classifieurs, similarité inter-concepts, ontologie, LSCOM-lite, TRECVid.

Abstract

Today, the access to documents in databases, archives and Internet is mainly through textual data : image names or keywords. This search is not without faults : spelling, omission, etc. The recent advances in the field of image analysis and machine learning could provide solutions such as features-based indexing and retrieval, using color, shape, texture, motion, audio and text. These features are rich in information, especially from the semantic point of view.

This work deals with information retrieval and aims at semantic indexing of multimedia documents : video shots and key-frames. Indexing is an operation that consists of extracting, representing and organizing the content of documents in a database.

However, indexation is confronted with the "semantic gap" problem between low-level visual representations and high-level features (concepts). To limit the consequences of this issue, we introduced into the system, different types of descriptors, while taking advantage of the scientific advances in the field of machine learning and the multi-level fusion. Indeed, fusion is used to combine several heterogeneous information from multiple sources, to obtain more complete, global and higher quality information. It can be applied to different levels of the classification process. Here, we studied the low-level feature fusion, high-level feature fusion and decision fusion.

First, we present a state of the art of high-level fusion methods, in the indexing and search systems. In particular, the adaptation of evidence theory to neural network, thus giving *Neural Network based on Evidence Theory (NNET)*. This theory presents two important information for decision-making, compared to the probabilistic methods : belief degree and system ignorance. Then, NNET has been improved by incorporating the relationship between descriptors and concepts, modeled by a weight vector based on entropy and perplexity. The combination of this vector with the classifiers outputs, gives us a new model called *Perplexity based Evidential Neural Network (PENN)*.

We have also introduced the important topic of ontology and inter-concepts similarity (i.e. the study of relations between the classes). Indeed, the concepts are not generally expressed in isolation and a strong correlation exists between certain classes. The first difficulty lies in the use of an ontology that describes the relationships between concepts. The second concerns us more, is the operation of this semantic information. Three types of information are used : low-level visual descriptors, co-occurrence and semantic similarities, in conjunction with a multimedia knowledge database for semantic interpretation of video shots. The final system is called *Ontological PENN*.

Finally, we respond to the question concerning the usefulness of the low-level fusion. This was possible only through a statistical study of data before and after features fusion. The proposed systems have been validated on data from TRECVid (NoE K-Space project) and soccer videos provided by Orange-France Telecom Labs (CRE- Fusion project).

Keywords : Video shots indexing, semantic gap, classification, feature fusion, classifier fusion, inter-concepts similarity, ontology, LSCOM-lite, TRECVid.

Remerciements

La thèse résulte d'un travail personnel dont l'aboutissement implique à différents degrés de nombreuses personnes. Je prends donc le temps de remercier les acteurs du bon déroulement de ces trois années et demi de doctorat.

Ce travail n'aurait pas vu le jour sans mon encadrant Benoit Huet, qui m'a offert la possibilité de travailler dans un domaine de recherche original et fort intéressant. *"Je souhaite t'exprimer ma gratitude pour ta disponibilité associée à ton impressionnant dynamisme scientifique tout au long de ces années de thèse.* Je suis également reconnaissant à Sid-Ahmed Berrani, ingénieur-chercheur chez Orange-France Télécom de Rennes, qui a grandement participé à l'encadrement de mes recherches.

J'ai déjà pu remercier les membres de mon jury mais ils méritent une redite. Je remercie donc Hervé Glotin et Philippe Mulhem d'avoir accepté d'être les rapporteurs de mon mémoire. Je remercie les professeurs Gaël Richard, Peter Sander et Bernard Mérialdo d'être les examinateurs, ainsi que Sid-Ahmed Berrani d'être membre invité.

La suite des remerciements est destinée à mon entourage. Merci à tous mes amis Sophipolitains d'avoir été là, en particulier Christophe Chevallier : *"Ta bonne humeur m'a permis de surmonter mes difficultés, ne change rien, je suis heureux de t'avoir connu et surtout ne perdons pas contact !"*. Mes collèges Marco, Eric, Simon, Taoufik et Ahcene m'ont toujours apporté une grande quiétude propice à la réflexion. Sans oublier le personnel d'Eurécom.

Merci à mes parents *"Merci Yema"* et à ma magnifique famille de m'avoir toujours encouragé et aidé à mener mes études si longtemps. Leur soutien constant à travers ces longues années m'a été précieux.

Pour terminer, je remercie Coralie Fievet pour son sourire inaltérable, sa présence et son attention quotidienne. *"Merci Coralie et à ton stylo rouge d'avoir bien voulu relire et corriger mon mémoire, de m'avoir accompagné et encouragé durant des moments pas toujours faciles. J'espère que nous pourrons rapidement concrétiser nos rêves"*.

Les derniers remerciements vont à ceux que je n'ai pas cités et qui pourtant le méritent. Je suis sûr qu'ils se reconnaîtront et c'est le principal.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

BKS	Behavior knowledge space	Espace de connaissance du comportement
CBIR	Content-based image retrieval	Recherche d'images par le contenu
CBVR	Content-based video retrieval	Recherche de vidéos par le contenu
DT	Decision template	
DCT	Discreet cosine transform	Transformée discrète en cosinus
GMM	Gaussian mixture model	Mélange de gaussiennes
IVSM	Image vector space model	Modèle d'espace vectoriel appliqué aux images
KNN	K-nearest neighbors	K-plus proches voisins
LDA	Linear discriminent analysis	Analyse discriminante linéaire
LSA	Latent semantic analysis	Analyse de la sémantique latente
MPEG	Moving picture expert's group	
NMF	Non-negative matrix factorisation	Factorisation en matrices non-négatives
NN	Neural network	Réseau de neurones
NNC	Neural network Coder	Codage par réseau de neurones
NNET	Neural network based on evidence theory	Réseau de neurone basé sur la théorie des évidences
PENN	Perplexity-based evidential neural network	
PCA	Principal components analysis	Analyse en composantes principales
RBF	Radial basis function	Fonction à base radiale
SVD	Singular value decomposition	Décomposition en valeurs singulières
SVM	Support vector machine	Machine à vecteurs de support
WBF	Weighted BagFolding	

Introduction Générale

« Une image vaut mille mots. »
Confucius.

Motivations

Grâce aux progrès technologiques de ces dernières années, en particulier dans les télécommunications et l'informatique, l'information numérique est devenue le coeur de nos systèmes socio-économiques [Jol00]. Ces progrès se sont accompagnés d'une baisse des coûts qui a facilité la diffusion vers le grand public. Cependant, l'information numérique n'a pas seulement pour objectif de créer un espace de consommation. Dans plusieurs cas, nous souhaiterons la réutiliser et si possible sur le long terme. Ceci, pose la problématique de la pérennité et de la résistance des espaces de stockage. En effet, la principale préoccupation des chercheurs était la compression de nombreux documents, qui ne se limitent plus au texte mais qui incluent aussi l'image et la vidéo. Aujourd'hui, la question qui se pose est la suivante : comment gérer cette masse d'informations pour y accéder efficacement, en prenant en compte l'infinité de documents numérisés disponibles sur Internet [1], le câble et sur les bouquets de télévision par satellite, en Pay Per View, ou encore en vidéo à la demande (VOD) ?

	Janv	Fev	Mars	Avr	Variation (%)
1 To	366.56	353.53	340.25	322.77	-11.94
750 Go	229.02	215.64	204.09	195.39	-14.68
500 Go	165.70	163.89	160.63	155.54	-6.13
320 Go	126.48	127.87	125.73	121.36	-4.04
250 Go	120.24	116.01	112.41	106.45	-11.46
160 Go	96.04	93.33	91.93	89.58	-6.72
80 Go	76.80	73.83	71.82	70.01	-8.84

TAB. 1 – Le site *www.touslesprix.com* présente une étude sur l'évolution du prix (en Euro) des disques durs sur le marché pour les 4 premiers mois de l'année 2008.

L'usager de demain ne se contentera plus de rechercher par mots-clés dans une banque

[1] Selon comScore datant du 20 août 2008, 81% des 25 millions d'internautes français ont visualisé 2.3 milliards de vidéos en ligne, pendant le mois de mai 2008.

de documents. Il voudra, par exemple, filtrer les reportages sportifs de son bulletin télévisé, retrouver une séquence d'un film dans ses archives, écouter un titre musical en fredonnant quelques notes du refrain, ou encore, un meilleur contrôle parental de son navigateur Internet pour bloquer l'accès ou l'apparition de tout document à contenu offensant pour ses enfants. Par ailleurs, le développement des sites communautaires et les réseaux sociaux sur Internet, la téléphonie-mobile et les appareils photos a produit une numérisation de notre environnement quotidien, ce qui explique l'engouement des chercheurs et des industriels pour l'imagerie numérique.

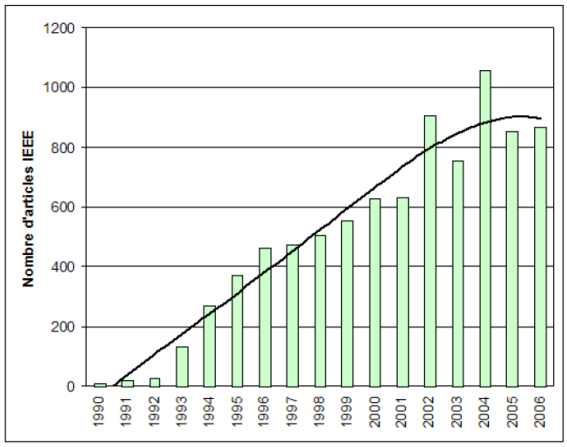


FIG. 1 – Nombre d'articles IEEE référençant la classification et la recherche d'images [Lef07].

Pour répondre à ces besoins, une meilleure appréhension du contenu visuel est soulevée, à travers l'indexation et la recherche par le contenu sémantique, moyennant des méthodes de classification pour donner un sens au contenu des documents, après une phase importante d'apprentissage. Il convient alors de reconnaître les caractéristiques du document telles que la couleur, la forme, la texture, le mouvement, le son et le texte, afin d'identifier leurs classes d'appartenance. Cependant, il est difficile d'appréhender le sens d'un concept pour constituer des classes, du fait de sa sensibilité à la taille, la résolution, l'illumination et aux conditions de prise de vue, etc. Un concept peut représenter une action, un objet, un lieu, ... Typiquement, les concepts PRISONNIER et PERSONNE présentent un certain niveau d'abstraction et une telle différence suscite l'usage de méthodes spécialisées et adaptées à ce genre de concepts. Ceci ne semble pas être la bonne direction à suivre vu le nombre de concepts et la question de l'exploitation de données de bas-niveau avec ceux du haut-niveau. Par ailleurs, face à la présence de bruit, aux spécificités et aux imperfections des capteurs, à la difficulté de représenter correctement le contenu, il est devenu très difficile d'améliorer les performances des outils de classification existants. La tendance actuelle est alors de mieux exploiter l'ensemble de l'information disponible en faisant appel aux systèmes de fusion et éventuellement, à de nouvelles sources d'informations. L'objectif est de créer un modèle générique, stable et robuste, permettant l'identification automatique du contenu des images et des plans vidéos à travers un apprentissage sur les informations pertinentes et l'introduction de la fusion multi-niveaux dans le processus.

En effet, la fusion est utilisée dans le but de combiner des informations hétérogènes issues de plusieurs sources afin d'obtenir une information globale, plus complète, de meilleure qualité, permettant de mieux décider et d'agir. Elle peut être appliquée sur plusieurs niveaux du processus de classification. Dans cette thèse, nous avons étudié la fusion de descripteurs bas-niveau dite fusion précoce *"feature fusion"* (i.e. les signaux émis par les capteurs sont combinés avant la classification), la fusion de descripteurs haut-niveau appelée aussi fusion tardive *"classifier fusion"* (i.e. chaque système de classification fournit son opinion, puis ces dernières sont combinées) et décisionnelle *"decision fusion"* (i.e. introduit la structure de l'ontologie et de la similarité inter-concepts dans le raisonnement). Nous allons exposer à présent les principales difficultés rencontrées lors de la mise au point d'un modèle, ce qui nous permettra de motiver la fusion.

Motivations de la fusion

La première difficulté réside dans le choix des descripteurs (caractéristiques) et de la multiplicité de leurs représentations. Par exemple, le contenu d'un document peut être décrit par des histogrammes de couleurs dans différents espaces (RGB, HSV, ...), par les textures le composant, par ses caractéristiques géométriques (points, coins, droites, ...), ou par d'autres descripteurs.

La deuxième difficulté est le choix du modèle. Souvent l'observation des mesures et de leurs sorties peut donner une idée du type de modèle à employer, mais sans aucune garantie qu'il se comportera mieux qu'un autre semblant moins adapté. Par exemple, nous avons une distribution des données qui suit une loi gaussienne. Dans ce cas, une modélisation par des mixtures de gaussiennes semble un choix judicieux. Toutefois, il est possible qu'un réseau de neurones soit plus performant ou que ces modèles soient complémentaires. De plus, il faut noter que la visualisation des mesures est loin d'être triviale étant donné la dimension de l'espace associé, ce qui rend le choix à priori du modèle encore plus délicat.

La troisième difficulté est le choix des paramètres du modèle ou de l'ensemble d'entraînement $\mathcal{L}$. Chaque modèle est associé à un ensemble d'algorithmes d'apprentissage, permettant l'estimation des paramètres. Ces derniers dépendent d'une phase d'initialisation. Donc, il existe plusieurs hypothèses [2] (ou ensemble de paramètres) qui répondent au problème posé. Il est à noter que la fonction optimale recherchée peut ne pas faire parti des hypothèses explorées, due aux limitations du modèle, à l'algorithme d'apprentissage ou à la taille et la qualité de $\mathcal{L}$.

Ces difficultés mettent en avant la nécessité de mettre en place un système de fusion. En effet, la construction d'une hypothèse nécessite de nombreux choix critiques qui ont tous leurs influences. Toutefois, l'ensemble des hypothèses possibles est très vaste et il n'est pas concevable de toutes les calculer. Dans la prochaine partie, nous allons voir comment le sujet est traité et les contributions qui sont apportées.

[2] L'entraînement consiste à déterminer les paramètres du modèle à partir de $\mathcal{L}$. Le modèle obtenu est alors appelé une hypothèse.

Nos contributions

Nous allons brièvement présenter les travaux réalisés dans le cadre de cette thèse ainsi que les contributions apportées, soit pour répondre à une difficulté rencontrée dans le traitement des données ou dans la conception du système d'indexation et de la recherche par le contenu. Nous proposons :

1. Une nouvelle technique d'apprentissage basée sur le principe du Bagging, l'Adaboost et le Ten-Folding appelée **Weighted BagFolding "WBF"**. Cette technique nous a permis de résoudre les problèmes liés aux sur-apprentissages, et le manque de données d'entraînement dans les bases de grande taille. Par ailleurs, un état de l'art et une dichotomie ont été introduits pour les méthodes de fusion de classifieurs (fusion haut-niveau). Des expériences sur les performances de ces méthodes ont été réalisées et publiées dans [BH06].
2. Un algorithme de fusion basé sur la théorie des évidences via la règle de combinaison de Dempster-Shafer, appelée **Neural Network based on Evidence Theory "NNET"** [BH07], pour la fusion de classifieurs. Cette méthode à l'avantage de présenter deux nouvelles informations importantes pour la prise de décision comparant aux méthodes probabilistes : l'ignorance du système et le degré de croyance dans une classe. Les expérimentations ont été conduites dans les projets NoE K-Space avec les données TRECVid et CRE-Fusion avec les données d'Orange-France Télécom Labs.
3. Par ailleurs, chaque concept du LSCOM-lite (Large-Scale Concept Ontology for Multimedia) [NKK+05] est mieux représenté ou décrit par un certain ensemble de descripteurs. Intuitivement, les descripteurs de couleurs peuvent être plus discriminant pour des concepts tels que SKY, SNOW, WATERSCAPE, VEGETATION, et moins discriminant pour STUDIO et MEETING. On propose de pondérer chaque descripteur bas-niveau selon son degré de discriminance par rapport aux concepts. Une solution à ce problème nous renvoie à l'utilisation de l'entropie. Elle mesure la quantité de l'information et d'incertitude dans une distribution. On propose de projeter les descripteurs visuels dans un vecteur poids via la mesure de l'entropie et de la perplexité. Ce vecteur est ensuite combiné avec les sorties des classifieurs pour produire une nouvelle entrée qui prend en compte la relation descripteur/concept dans le NNET. La combinaison des deux processus nous donne ce que nous appelons **Perplexity-based Evidential Neural Network "PENN"** [BH08b].
4. Pour la fusion de descripteurs bas-niveau, les méthodes statiques et dynamiques ont été étudiées, ainsi qu'un modèle de réseau multi-couches appelé **Neural Network Coder "NNC"** [BH08a] pour le codage de l'information, dans le but de réduire le nombre de dimensions. Une étude statistique et comparative des données (descripteurs) avant et après la combinaison, avec l'Analyse en Composantes Principales (ACP) a été effectuée.
5. Enfin, nous avons travaillé sur le thème important des ontologies et de la similarité inter-concepts. C'est à dire l'étude des relations entre les classes. En effet, les concepts ne sont pas exprimés de manière isolée et une forte corrélation existe entre certaines

classes. Une première difficulté est dans l'utilisation d'une ontologie qui décrit les relations existantes entre les concepts. La deuxième difficulté qui nous intéresse plus, réside dans l'exploitation de cette information sémantique et à priori par les systèmes de classification ou de fusion. Ces travaux ouvrent des perspectives intéressantes car ils offrent une passerelle entre une description de bas-niveau du plan par des descripteurs visuels et une description de haut-niveau, permettant l'extraction du contenu à haute valeur sémantique et par conséquent l'interprétation globale du plan vidéo [BH09a]. Pour cela, une étude de la similarité inter-concepts est effectuée en introduisant 3 types d'informations : les descripteurs bas-niveau, la cooccurrence et la similarité sémantique issue de l'approche hybride [BH09b]. Le système final s'appellera **Ontological PENN "Onto-PENN"**.

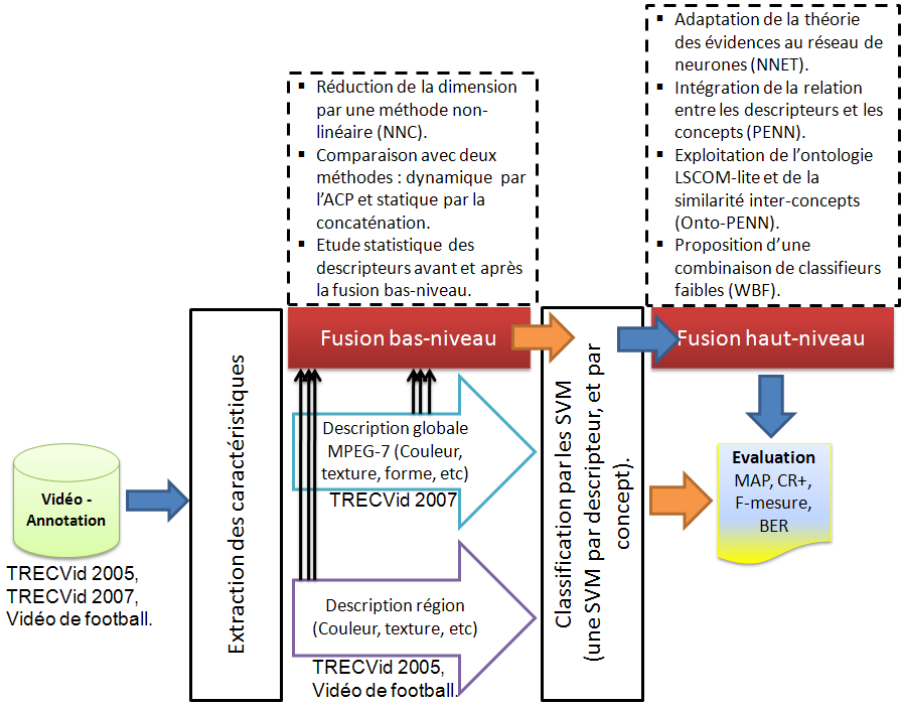


FIG. 2 – Représentation générale du processus d'indexation avec les contributions apportées dans la fusion multi-niveaux.

Plan de thèse

Le premier chapitre énonce la problématique générale de la thèse et les travaux effectués dans la littérature sur la modélisation d'un système de recherche et d'indexation par le contenu sémantique. Puis, il aborde la description globale de l'information vidéo, et décrit les données sur lesquelles nous allons travailler.

Le second chapitre expose un état de l'art sur les différentes méthodes d'extraction des

descripteurs, dont ceux considérés par le standard MPEG-7. Ensuite, il présente l'apprentissage automatique à travers la classification sémantique des plans vidéo. Pour cela, trois méthodes seront détaillées pour estimer les concepts sémantiques présents.

Le troisième chapitre s'intéresse à la multimodalité [3] en étudiant les différents aspects considérés quand on utilise plusieurs types d'informations. D'abord, nous présenterons la fusion de descripteurs haut-niveau. Une comparaison des méthodes décrites sera effectuée pour estimer les concepts sémantiques présents dans les plans vidéo. Nous proposons une nouvelle méthode NNET modélisée par la théorie des évidences et adaptée au réseau de neurones, puis, une extension de la méthode vers le PENN. Ensuite, nous traiterons de la fusion de descripteurs bas-niveau, essentiellement par deux approches : la fusion et la sélection. On se focalisa sur la première approche via la réduction du nombre de dimensions. Nous introduisons le NNC. Une comparaison avec une autre méthode de la même catégorie sera fournie.

Dans le quatrième chapitre, nous aborderons l'apport de la structure de l'ontologie et de la similarité inter-concepts dans notre système d'indexation appelé Onto-PENN en conjonction avec une base de connaissance multimédia pour l'interprétation sémantique de plans vidéo. Plusieurs constructions de la similarité inter-concepts seront étudiées.

Enfin, la conclusion générale présente une synthèse des travaux menés dans cette thèse ainsi que les perspectives liées à ce travail.

[3] La multimodalité désigne la présence de plusieurs modes (e.g. image, audio, texte dans un document vidéo). Dans un tel cas, la couleur, la texture et la forme dans une image sont plutôt des sous-modalités. Pour simplifier, nous étendons la notion de modalité à une information issue d'une source (capteur physique) ou d'un algorithme d'extraction de caractéristiques. Nous y aborderons essentiellement la représentation visuelle dans les vidéos, tout en soulignant l'usage du flux audio pour les campagnes d'évaluations, justifiant ainsi la notion du multimédia.

Chapitre 1

Analyse Multimédia

Avec le succès du multimédia à travers la télévision, sur nos téléphones, nos écrans d'ordinateurs et internet, la quantité de données ne cesse d'augmenter, profitant des tous derniers progrès technologiques effectués en compression, transmission et en stockage pour faire naître un phénomène de culture de l'image et qui occupe de plus en plus de place dans l'espace de la communication. Alors, la question qui se pose est de savoir comment retrouver un document qui nous intéresse dans cette grande masse de données ?

L'engouement croissant des chercheurs dans le domaine du ***Intelligent Multimedia Information Retrieval*** *traduit bien les enjeux de tels systèmes. Aujourd'hui, pour beaucoup de personne Google* [1] *est la référence dans la recherche de documents sur internet, la raison est qu'il arrive dans cet océan d'informations à retrouver ce que les utilisateurs recherchent, en affichant sans gène qu'il peut présenter des erreurs, comme le montre la Fig 1.1. En effet, le moteur de recherche est mis à défaut pour plusieurs images lorsqu'on veut visualiser des images de bateaux "boat" par exemple : on retrouve une image d'un lit d'hôtel qui porte le nom de "Boat Hotel", où d'un hamburger "boat marsin burger", où encore un plan représentant le "boat diner Paris", etc. Or, Google cherche que sur une petite partie de cet océan en s'appuyant exclusivement sur le texte à partir de mots-clés, sans exploiter l'information multimédia (images, vidéos, sons, etc). C'est précisément là que nos travaux de recherche commencent. Il est intéressant de noter que la recherche du même concept "boat" en utilisant une autre langue, donnera des résultats complètement différents.*

Dans ce chapitre, nous allons dans un premier temps introduire et comprendre le problème du fossé sémantique, pour ensuite mieux décrire la structure générale d'un système de recherche d'information par le contenu, suivi d'une présentation des techniques d'indexation d'image et de vidéo, puis, de la description du support vidéo. Enfin, nous allons présenter les données utilisées dans cette thèse, en particulier dans deux projets (NoE K-Space et CRE-Fusion), pour mieux comprendre le but de nos travaux et d'avancer les problèmes auxquels nous pourrions être confrontés.

[1] www.google.fr

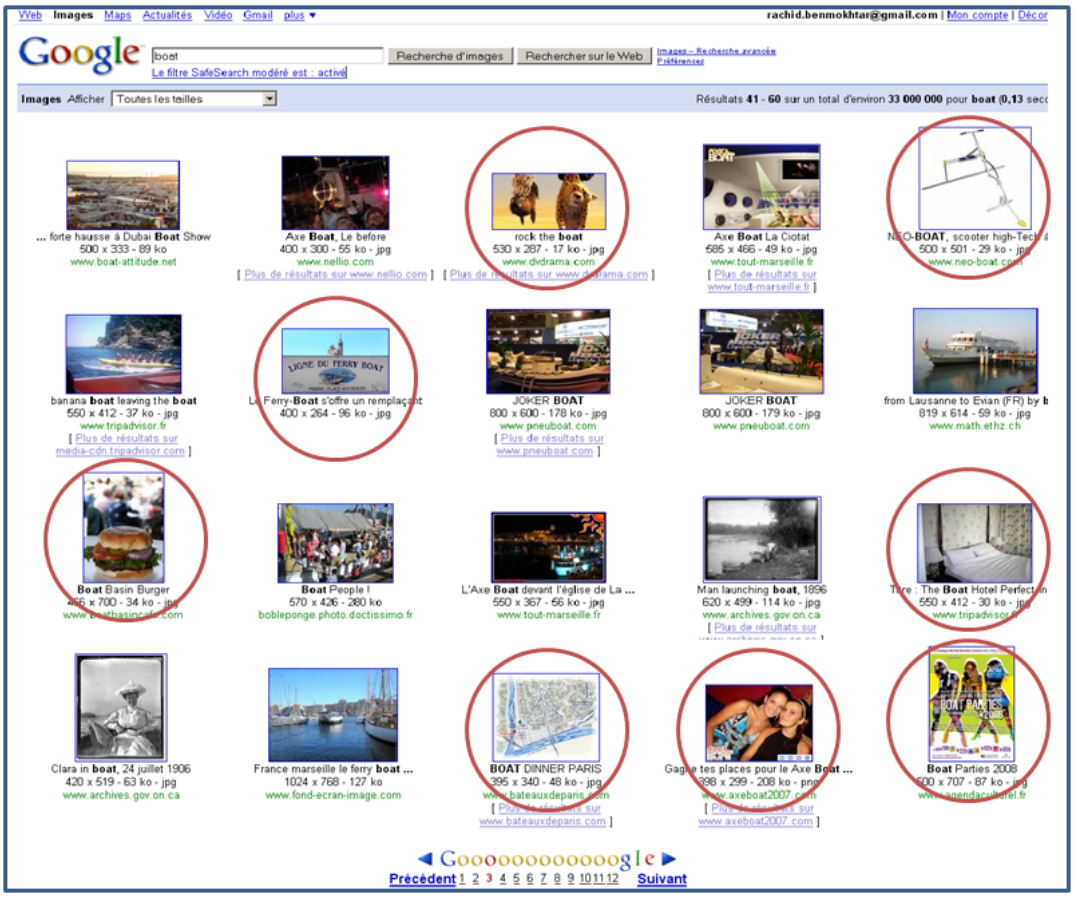


FIG. 1.1 – Résultats d'une recherche d'images par mots-clés pour le concept "Boat", sous le moteur de recherche Google. Les cercles rouges entourent les images qui ne représentent pas le concept "Boat" (i.e. les erreurs).

1.1 Le fossé sémantique

Si nous, humains savons parfaitement interpréter et détecter le contenu d'une image ou d'une information, nous sommes limités lorsqu'il s'agit de traiter une grande quantité d'informations. Un comportement inverse de celui de la machine, qui ne trouve pas de problème à traiter une tache répétitive donnée, mais limitée lorsqu'il s'agit d'interpréter ou d'extraire automatiquement des concepts à partir de données numériques, ce qui est appelé "fossé sémantique", définit comme suit :

1. **Smeulders et al.** [SWA+00] *The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation* [2].

[2] Le fossé sémantique est le manque de concordance entre les informations que la machine peut extraire d'un document numérique et des interprétations humaines.

2. **Ayache** [Aya07] *Le fossé sémantique est ce qui sépare les représentations brutes (tableaux de nombres) et sémantiques (concepts et relations) d'un document numérique.*

FIG. 1.2 – Les images (1) et (2) possèdent des histogrammes de couleurs très proches, contrairement aux images (2) et (3) qui sont différents. Les images sont extraites d'internet.

Pour comprendre cette difficulté, la Fig. 1.2 présente des images (1) et (2) qui possèdent des histogrammes [3] de couleurs très proches alors qu'elles n'ont aucun rapport sémantique. Par contre, les images (2) et (3) qui sont sémantiquement similaires (elles présentent toutes les deux une voiture), possèdent des histogrammes de couleurs très différents. Afin de limiter les conséquences de cette problématique sur le processus d'indexation, nous pensons que l'introduction de nouveaux et différents types de descripteurs bas-niveau dans le système, ainsi que l'utilisation des avancées scientifiques dans le domaine de l'apprentissage automatique en s'appuyant ou non sur l'utilisateur (*user-feedback*), et la prise en compte de la "fusion multi-niveaux" peuvent contribuer efficacement dans la création de relation, de rapprochement pour combler ce fossé entre les descripteurs bas-niveau avec ceux du haut-niveau. Enfin, la Fig. 1.3 expose quelques exemples de difficultés comme [Lef07] : la multitude de représentations, la variation d'angle de prise de vue, le changement d'échelle, la variation de luminosité, l'arrière plan différent, image de mauvaise qualité, etc.

La Fig. 1.3.a montre une première difficulté pour la modélisation du concept visuel "PAPILLON" présentant plusieurs espèces. Les méthodes de classification automatique se heurtent à cette multitude de représentations. Cependant, la recherche de caractéristiques communes (e.g. la forme et les contours des ailes du papillon) peut être la clé de ce problème. Une autre difficulté est celle des variations d'angle de prise de vue d'une VOITURE, illustrée par la Fig. 1.3.b. Dans la Fig. 1.3.c, on montre que la complexité varie avec la taille de l'objet recherché "VÉLO" (e.g. la complexité est grande si l'objet fait 5% de l'image que s'il est représenté en gros plan). Enfin, la Fig. 1.3.d présente une 4ème difficulté qui est le changement d'illumination, cette dernière est inévitable si les images sont prises à des heures différentes de la journée, à l'intérieur ou à l'extérieur d'une pièce. La présence d'ombre ou de halos sur un objet modifie le contenu visuel de l'objet et provoque parfois des occultations partielles.

[3] Un histogramme est une estimation de la densité de probabilité. L'histogramme de couleur est construit en deux phases. La première est la quantification des couleurs. La deuxième est le dénombrement des couleurs quantifiées, qui constituera l'histogramme. Chaque composante du vecteur de dénombrement donne la quantité d'une couleur présente dans l'image. Les histogrammes sont souvent normalisés par le nombre de pixels pour les rendre invariant à la taille de l'image ou des régions.

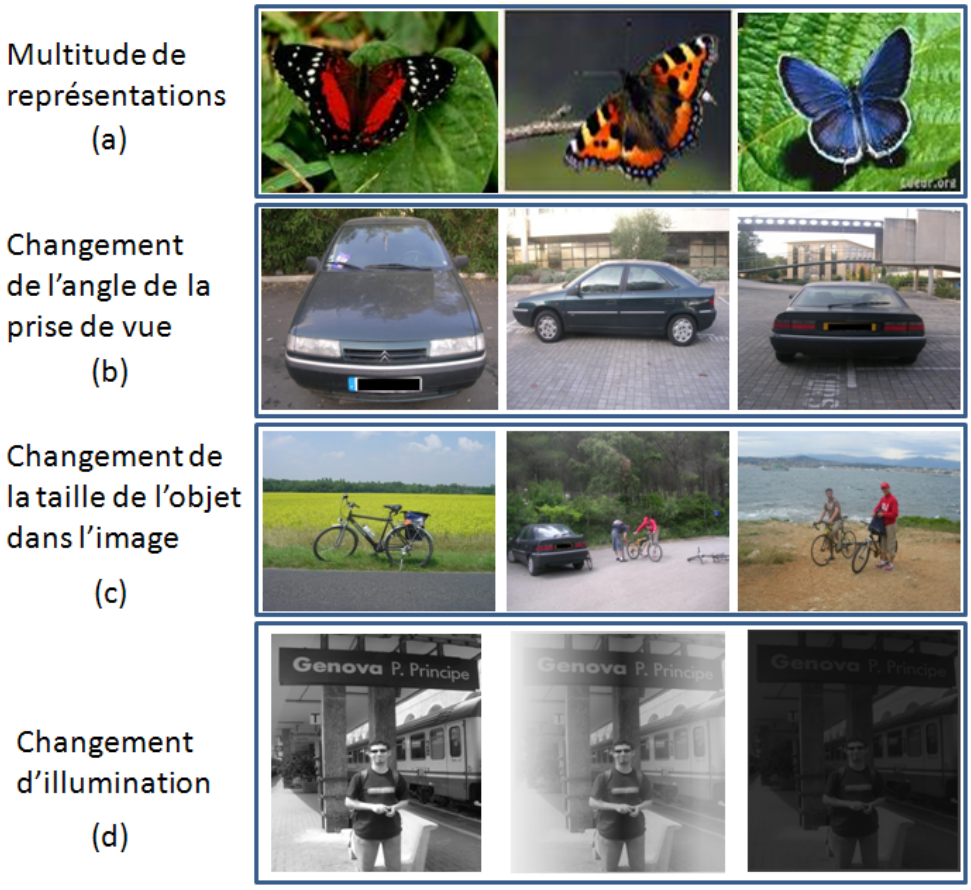


FIG. 1.3 – Exemples de difficultés pour l'indexation : la multitude de représentations, la variation d'angle de prise de vue, le changement d'échelle, la variation de luminosité. Les images de papillons sont prises sur internet, le reste appartiennent à ma base d'images personnelle.

1.2 Description générale du système

Les systèmes de recherche d'informations "Content Based Information Retrieval (CBIR)" basés sur le contenu présentent deux phases : l'indexation et la recherche. Le CBIR a pour but de satisfaire les besoins d'un utilisateur en retournant les documents les plus pertinents. La Fig. 1.4 illustre l'architecture générale d'un CBIR (e.g. à partir de vidéos) présentée dans ce qui va suivre.

1.2.1 Indexation

L'indexation consiste à extraire, représenter et organiser efficacement le contenu des documents d'une base de données. Pour cela, les documents sont tout d'abord représentés par une signature numérique réalisée par les deux étapes suivantes, ce qui permettra leur

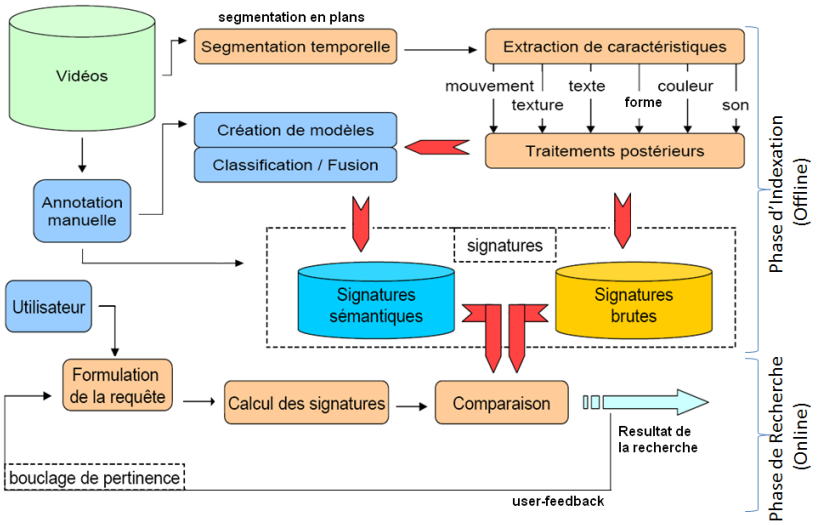


FIG. 1.4 – Composantes du CBIR, avec deux phases : Indexation et Recherche [Sou05].

identification :

1. La première étape implique l'extraction des caractéristiques d'un document, à travers la capture des couleurs, des textures, des formes et du mouvement dans un plan, etc.
2. La seconde étape permet de compresser l'information extraite tout en conservant l'essentiel. Il est important d'avoir des signatures compactes pour éviter d'avoir des données trop importantes à stocker et à traiter. A ce stade, nous avons des *signatures brutes*. D'autres types de signatures sont obtenues en détectant automatiquement des concepts de plus haut-niveau qu'on appelle *signatures sémantiques*. Plus de détails sur cette dernière seront donnés dans le chapitre 2.

1.2.2 Recherche

La recherche d'informations consiste en un ensemble d'opérations pour répondre à la requête d'un utilisateur par l'intermédiaire d'une interface utilisateur. En règle générale, la difficulté est d'exprimer correctement l'objet de la requête en utilisant au mieux les moyens proposés par le système. Cette requête doit, bien entendu, être comprise par ce dernier. Il existe plusieurs types de requête [VD00] :

- **recherche par mots-clés :** l'utilisateur donne des mots sensés représenter l'image recherchée. Souvent, il dispose d'une série de termes prédéfinis pour formuler sa requête.
- **parcours au hasard :** la base est parcourue aléatoirement jusqu'à ce que l'utilisateur trouve l'image qui l'intéresse.
- **navigation par catégorie :** les images de la base sont classées par catégories. L'utilisateur peut donc directement choisir la catégorie dans laquelle il pense pouvoir trouver l'image.

– **recherche par l'exemple :** l'utilisateur fournit une image exemple et le logiciel recherche dans la base, les images qui ressemblent à l'image exemple. Cette dernière peut être une photo de l'objet désiré ou une représentation créée par l'utilisateur lui même (dessin ou image de synthèse). C'est plutôt ce dernier type de requête qui sera retenu dans cette thèse.

Ensuite, la requête est transformée en signature utilisant le même procédé que celui de l'indexation. La suite consiste généralement à définir des distances/mesures de similarité globales entre les signatures/images. On peut alors calculer l'ensemble des mesures de similarité entre l'image requête et ceux de la base pour retrouver l'information la plus pertinente. Il suffit d'ordonner les images de la base suivant leur score et de présenter le résultat à l'utilisateur. Les images ayant le plus grand score de similarité étant considérées comme les plus proches.

Toutefois, il est difficile de répondre aux exigences des utilisateurs à partir d'une seule requête. Il est alors utile d'intégrer un bouclage de pertinence (user-feedback) incluant l'avis de l'utilisateur pour améliorer la requête en fonction du résultat précédemment obtenu, ce qui permet également à l'utilisateur de clarifier sa demande qui est souvent mal formulée.

1.2.3 Contexte national et international

Il existe plusieurs prototypes implémentant les systèmes d'indexation et de recherche par le contenu. Le domaine fait toutefois encore parti de la recherche et n'est pas encore mature aussi bien au niveau national qu'au niveau international. Dans ce qui va suivre, on citera dans la Table. 1.1 quelques exemples de projets afin de montrer l'intérêt grandissant de ce domaine, même s'il est clair que l'indexation par le contenu sémantique est une interaction entre plusieurs compétences (e.g. apprentissage automatique, base de données, intelligence artificielle, interaction homme/machine, traitement de l'information, etc), ce qui a favorisé plusieurs collaborations au sein de divers projets.

Au niveau national, une communauté structurée autour de groupes de recherche a vu le jour à travers le GdR-PRC I^3 (Information, Interaction, Intelligence), elle s'intéresse aux aspects de l'apprentissage automatique, l'intelligence artificielle, l'interaction homme machine et les bases de données, ainsi que le GdR-PRC ISIS (Information, Signal, Images et viSion), qui s'intéresse aux aspects traitement statistique de l'information, du moins pour ce qui nous concerne. Ceci a conduit au financement de plusieurs projets et la création de deux pôles de compétitivités *Image & Réseau* dans la région de Bretagne et *CapDigital* dans la région Parisienne, où l'on trouve le projet Infom@gic qui a pour ambition de créer une plate forme couvrant l'aspect moteur de recherche, l'extraction des connaissances et la fusion d'informations multimédia.

1.2.4 Systèmes de recherche par le contenu existant

Si on regarde la colonne du budget dans la Table. 1.1 des projets de recherche concernant les systèmes d'informations de ces dernières années, les industriels comme les pouvoirs publics ont compris la place stratégique de ce secteur dans leur système sociétal et économique. En effet, l'indexation et la recherche d'image par le contenu sont devenues des

Projet	Durée	Budget (MEuro)	Financement
PASCAL (Pattern Analysis Statistical Modeling and Computational Learning)	2003 à 2007	5.44	Union européenne
DELOS (Network of Excellence on Digital Libraries)	2004 à 2007	15.7	Union européenne
MUSCLE (Multimedia Understanding through Semantics, Computation and Learning)	2004 à 2008	6.9	Union européenne
K-SPACE (Knowledge Space of Semantic Inference for Automatic Annotation and Retrieval of Multimedia Content)	2006 à 2008	8.9	Union européenne
PHAROS	2007 à 2009	14.2	Union européenne
THESEUS	2007 à 2011	120	Allemagne
Europeana	2008 à 2010	120	Union européenne
QUAERO	2008 à 2013	250	France Allemagne

TAB. 1.1 – Exemples de projets de recherche dans l'indexation multimédia et la création de moteur de recherche.

pôles très actifs de la recherche. De nombreux systèmes commerciaux et académiques ont été proposés pour l'image, puis rapidement étendus à la vidéo.

Cependant, il est très difficile de comparer les résultats de ces produits, et cela pour plusieurs raisons : ils utilisent tous des bases d'images différentes, elles ne sont pas connues dans leur intégralité par l'utilisateur ce qui rend impossible de calculer une quelconque efficacité ou précision, ensuite, nous ne pouvons pas savoir si les bases ne sont pas déjà pré-triées (souvent les images similaires se suivent par leur numéro).

Dans la Table. 1.2, nous allons présenter brièvement les principaux systèmes d'indexation d'image et de la vidéo en montrant les évolutions et les améliorations apportées au fil du temps [RHC99, VD00].

1.3 Représentation de la vidéo

Suite à la présentation de l'évolution des systèmes d'indexation et de recherche par le contenu, cette section traitera du support numérique de la vidéo comme le montre la Fig. 1.5, sans s'attarder sur leur codage.

Systèmes	propriétés
QBIC [FBF+94]	Développé par IBM, Premier système commercial de recherche d'image par le contenu, La requête est formulée grâce à la recherche par l'exemple, Utilise la couleur, la texture, etc. http ://wwwqbic.almaden.ibm.com/
Virage [BFG+96]	Développé par Virage Inc, Similaire à QBIC, Permet la combinaison de plusieurs types de requêtes, L'utilisateur peut attribuer des poids pour chaque mode, http ://www.virage.com/
RetrievalWare [BHJ+05]	Développé par Excalibur Technologies Corp, Système à base de réseau de neurones, Utilise la couleur, la forme, la texture, la luminance et la localisation des couleurs et la structure de l'image, Permet la combinaison de tous ces modes avec des poids définis par l'utilisateur, http ://vrw.excalib.com/cgi-bin/sdk/cst/cst2.bat
Photobook [PPS94]	Développé par MIT Media Laboratory, Se base sur 3 critères (couleur, texture et forme), Utilise plusieurs méthodes (distance euclidienne, mahalanobis, divergence, histogrammes, vecteurs d'angle, etc) La version améliorée permet la combinaison de ces méthodes, http ://www-white.media.mit.edu/vismod/demos/photobook/
Netra [YM98]	Développé dans le projet UCSB Alexandria Digital Library, Utilise la couleur sur des régions pour chercher des régions similaires dans la base de données, La version 5 utilise le mouvement dans la segmentation spatio-temporelle, http ://vivaldi.ece.ucsb.edu/Netra
Informedia [WKSS96]	Développé par l'université Carnegie Mellon, Exploite le mouvement de la caméra et l'audio, Réalise une reconnaissance vocale automatique, http ://www.informedia.cs.cmu.edu/
SurfImage [NMB+98]	Développé par l'INRIA, Permet de combiner les caractéristiques bas et haut-niveau, L'utilisateur sélectionne une image et une mesure de similarité, et le poids de chaque caractéristique, http ://www-rocq.inria.fr/imedia/index-UK.html

TAB. 1.2 – Quelques exemples de systèmes d'indexations et de recherches d'images et de vidéos.

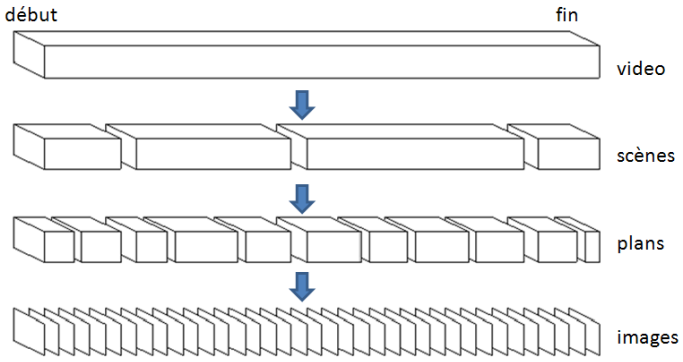


FIG. 1.5 – Structure d'une vidéo.

1.3.1 Segmentation en plans

Le plan constitue l'unité technique de base des systèmes de classification, d'indexation et de recherche par le contenu. Il est défini comme une séquence d'images prises par une caméra durant laquelle l'acquisition du signal est continue, souvent de courte durée. Un plan est identifié dans une zone limité de l'espace où se déroule l'action, par le processus de segmentation en plans, qui regroupe les images selon leurs caractéristiques bas-niveau ; où la différence calculée entre deux images conclura sur la présence ou non, d'un changement de plan. La littérature présente l'utilisation de nombreuses caractéristiques accompagnées de leur mesure de comparaison. Nous retrouvons principalement les caractéristiques suivantes : les histogrammes locaux ou globaux de couleurs [ZKS93] et de la structure (essentiellement contours et coins) [ZMM99]. Toutes ces méthodes reposent sur un seuil de différence, qui est sensible à la nature de la vidéo.

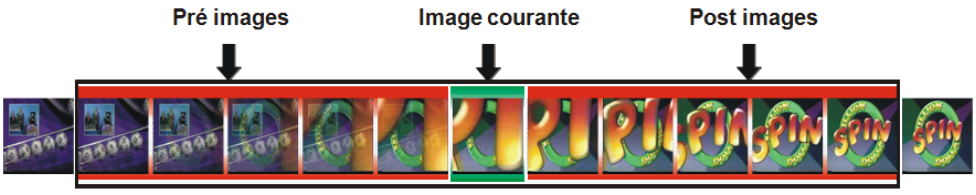


FIG. 1.6 – *Moving Query Window* de taille de 2x5 images (5 images en amont et 5 en aval de l'image courante) [VTW04].

Dans le cadre du projet CRE-Fusion [4] avec Orange-France Télécom Labs, nous avons utilisé l'algorithme de segmentation en plans *Moving Query Window* proposé par Volkmer et al. [VTW04] pour la détection des transitions franches "*Cuts*" et progressives. Cet algorithme est basé non pas sur l'étude de la similarité avec l'image précédente et la suivante, mais sur une fenêtre coulissante qui présente un certain nombre d'images en amont et en aval de l'image référence comme le montre la Fig. 1.6, afin de mieux prendre en considération

[4] Plus de détails sur le projet CRE-Fusion seront présentés dans la prochaine section.

l'évolution temporelle du plan vidéo. Cette implémentation accomplit simultanément les deux détections.

L'étude et l'analyse de nouveaux algorithmes de segmentation en plans n'entre pas dans le cadre de cette thèse. Nous allons donc brièvement présenter la méthode disponible que nous avons utilisé.

Pour la détection des transitions franches, chaque image est divisée en 16 blocs sans chevauchement, puis est extrait un histogramme HSV pour chacun. Il suffit de comparer la similarité entre les images de la fenêtre coulissante avec l'image courante à travers le calcul d'une distance Euclidienne par exemple, en donnant un poids à chaque bloc. Ensuite, on trie de façon décroissante les distances obtenues et on analyse ce classement. L'algorithme arrive à détecter une coupure lorsque le nombre d'images en aval classée dans la première partie du classement est supérieure à un seuil préfixé. Il a été observé qu'un mouvement rapide dans un plan peut fausser le résultat de la segmentation. Pour cela, l'auteur donne plus d'importance aux régions qui représentent le fond de l'image que les régions centrales. Ceci a permis l'obtention d'un gain de performance de 5% sur les données de TRECVid 2003, aussi, reduisant le temps de calcul [VTW04].

Pour les transitions progressives, les images qui précédent et suivent l'image courante sont regroupées dans deux ensembles d'images distinctes. La distance moyenne de chaque ensemble par rapport à l'image courante est calculée, en utilisant toutes les régions avec le même poids. Le rapport noté *PrePostRatio* entre la distance de l'ensemble des images en amont et celles en aval est calculée, pour indiquer la fin d'une transition progressive sous forme d'un pic dans la courbe *PrePostRatio vs images*.

L'application de cette méthode de segmentation en plans sur des vidéos de sport, représentant des matchs de football, a permis de relever les points suivants :

- la méthode de segmentation en plans est efficace pour les transitions franches,
- pauvre pour les transitions graduelles,
- dépend de l'espace de couleur,
- présente quelques difficultés lors de l'annotation due à la spécificité sémantique des classes choisis par Orange-France Télécom (e.g. l'attribution d'une classe à un plan qui présente une progression du ballon de la cage de gauche vers la cage de droite par exemple). Pour cela, il a été décidé conjointement avec Orange-France Télécom Labs d'orienter le projet vers un système de détection du contenu sémantique dans les images-clés des plans comme le montre la Fig. 1.7.

Enfin, nous citerons aussi en particulier, les systèmes de segmentation en plans présentés par [BSM+00, Que01] qui ont fourni pendant plusieurs années la segmentation officielle de TRECVid.

1.3.1.1 Sélection des images-clés dans un plan

Il est important de souligner qu'il est inutile d'utiliser toutes les images d'un plan pour obtenir une bonne extraction des caractéristiques visuelles. En effet, il serait par la suite impossible de conserver et d'utiliser cette information qui est par ailleurs redondante. Ainsi, une sélection d'une ou plusieurs images représentatives des plans serait nécessaire. Or, on peut déjà noter que si un plan présente de faibles mutations, il pourra être considéré comme

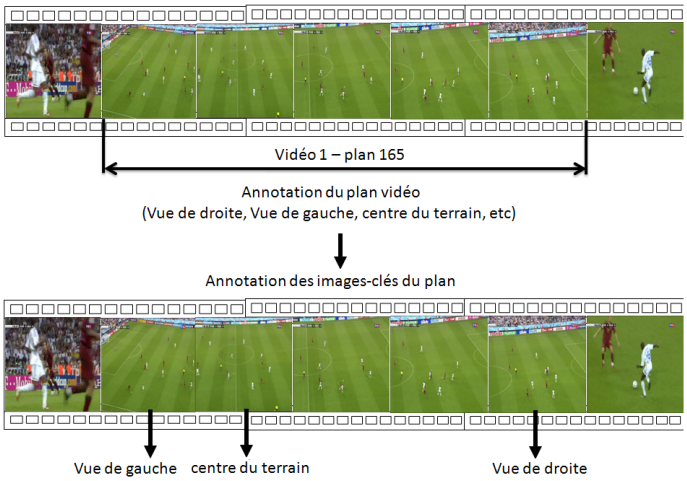


FIG. 1.7 – Difficulté lors de l'annotation d'un plan vidéo issu d'un match de football (vidéo : 1, plan : 165).

statique (e.g. un plan d'une présentatrice d'un journal télévisé), les images le composant sont très similaires. Il suffit alors de choisir l'image qui est la plus identique aux autres. En pratique, cette recherche exhaustive est difficilement réalisable. Les approches empiriques sélectionnent simplement la première, la dernière ou l'image médiane du plan [UMY91]. D'autres travaux comme ceux de : Yueting et al.[YYHM98] proposent une approche par regroupement des images similaires pour obtenir la ou les images représentatives du plan. Toutefois, lorsqu'un plan présente du mouvement, il est intéressant de sélectionner les images représentatives en fonction de l'intensité ou des variations du mouvement. Pour cela, Kobla et al.[KDLF97] utilisent le domaine compressé MPEG pour mesurer le déplacement de la caméra dans le plan et découpent ce dernier en sous plans afin de limiter l'amplitude du mouvement. Liu et al.[LZQ03] proposent une mesure de l'énergie du mouvement dans le domaine MPEG afin d'identifier les images-clés du plan, etc.

Au total entre 1 à 5 images-clés sont sélectionnées par plan pour le projet CRE-Fusion (i.e. plus la durée est longue plus on prend d'images pour les annotés). L'idée est de capturer le contenu de plans variés ou incorrectement segmenté.

1.3.1.2 Discussion

Il est clair que de nombreuses méthodes d'indexation et de recherche par le contenu visuel des vidéos sont très similaires aux méthodes employées sur les images puisqu'elles se concentrent uniquement sur les images-clés. Nous pensons qu'il est intéressant de nous situer rapidement par rapport aux différents travaux en relation avec le projet dans la détection des événements appliqués jusqu'à ce jour au sport.

Généralement, l'objectif est de trouver les moments forts d'un match dans le but de résumer une retransmission sportive de façon efficace. Ainsi, les coups-francs, les penaltys et les corners d'un match de football peuvent être détectés à l'aide d'un détecteur de mouvements

de caméra et d'un modèle de Markov caché (Assfalg et al. [ABB+02]). Yow et al. [YYYL95] s'intéressent à l'extraction d'événements dans un match de football et proposent de reconstruire une vue panoramique de l'action. Pour cela, ils détectent les cages sur le terrain, procèdent à un suivi du ballon et le mouvement est utilisé pour déterminer l'intensité de l'action. Cabasson et al. [CD02] combinent des règles basées sur des informations sonores et visuelles pour sélectionner les passages importants d'un match ; ils utilisent le mouvement pour détecter les actions importantes et l'énergie sonore pour détecter les réactions du public et les commentaires. Des travaux similaires ont été appliqués au basketball (Saur et al. [STKR97] ; Nepal et al. [NSR01]), au cricket (Lazarescu et al. [LVW02]), au football américain (Li et et al. [LS02] ; Babaguchi et al. [BKK02]), aux sports mécaniques (Petkovic et al. [PMJDK02]) et au baseball (Chang et al. [CHG02]).

Toutes les techniques citées sont spécifiques à un seul sport. D'autres travaux proposent une approche générale pour différents sports. Li et al. [LS02] détectent les actions importantes d'une vidéo de sport basée sur une approche à la fois déterministe et stochastique. Les expérimentations ont été conduites sur des vidéos de football, de baseball et de combats de sumos. Sadlier et al. [SO05] utilisent des caractéristiques comme la mesure de l'activité sonore, la détection de gros plans, de la foule et d'autres informations visuelles ou sonores sont extraites de la vidéo. Puis, une SVM est utilisée pour détecter les actions importantes parmi les actions candidates.

Notre travail dans le projet CRE-Fusion a pour objectif la conception et l'implémentation d'un système de détection d'un certain nombre de concepts particulier aux images de football et de voir le rôle de la fusion multi-niveaux dans ce genre de système. Nous pouvons nous placer dans la continuité des travaux de Assfalg et al. [ABB+02].

1.3.2 Segmentation en scènes

La scène constitue l'unité sémantique permettant d'exprimer une idée de l'action s'y déroulant ou de l'ambiance dégagée dans un moment de la vidéo. Elle est faite d'une séquence de plans souvent de courte durée. Toutefois, le problème de la segmentation en scènes est délicat, son utilisation pour l'indexation et la recherche d'information en est pour l'instant à ses débuts. En effet, deux catégories se distinguent : la première catégorie utilise les algorithmes de regroupement, où les plans sont regroupés en fonction de leur similarité [YY96]. La deuxième catégorie regroupe au fur et à mesure les plans sur la base d'un chevauchement des liens qui les relies en terme de contenus visuels et en variations temporelles. Un ensemble de règle basé sur l'application d'une fonction adaptative de seuil permettra de définir si un plan appartient à la scène courante ou à une nouvelle scène [HLB99, TZ04].

1.4 Présentation des données

Dans cette section, nous allons décrire les données utilisés dans le but de valider notre système d'indexation et de voir l'apport de la fusion dans les différentes étapes du processus. Au fur et à mesure de l'évolution de la thèse et de notre participation aux différents projets, trois bases de données ont été utilisées. Nous avons commencé par les données

TRECVid 2005 pour l'étude comparative de l'état de l'art de la classification et de la fusion de descripteurs haut-niveau dans le cadre du projet NoE K-Space [KSp]. Ensuite, les vidéos de football fournies par Orange-France Télécom Labs ont été introduites pour l'analyse de la fusion de descripteurs bas-niveau du projet CRE-Fusion. Enfin, nous emploierons les données TRECVid 2007 pour l'étude des relations descripteurs/concepts et de la similarité inter-concepts [5].

1.4.1 La campagne d'évaluation TRECVid

La campagne d'évaluation TRECVid [TRE] a pour objectif de favoriser la recherche scientifique, en particulier la recherche et l'indexation des documents multimédia par le contenu sémantique. Elle a permis aux participants la possibilité d'un véritable échange.

Nos travaux sont réalisés sur l'ensemble des vidéos fournies par TRECVid'05 à 2007 avec 40 concepts regroupés dans trois catégories [6] : évènement, scène et objets. Deux avantages sont à retenir de ce partage des annotations : (1) l'obtention d'une grande quantité de vidéos annotées, (2) la possibilité de comparer objectivement les différents systèmes proposés puisqu'ils sont entraînés et testés sur les mêmes vidéos. Notons que nos systèmes proposent des méthodes génériques à chaque fois, et ont obtenus de bonnes performances à la dernière évaluation (Benmokhtar et al. [BGH07]).

1.4.1.1 TRECVid 2005/2006

L'objectif que nous nous fixons est de construire des modèles permettant d'estimer les concepts choisis. Pour cela, nous allons utiliser dans un premier temps la base de données TRECVid'05 qui regroupe 85 heures de journaux télévisées en 3 langues : nouvelles américaines (canaux de MSNBC et CNN), nouvelles chinoises (canaux de NTDTV et PHOENIX) et nouvelles arabes (canaux de LBC et HURRA) avec leurs annotations. L'unité de traitement et d'évaluation est le plan. Le découpage des vidéos en plan est fourni par TRECVid. Nous avons réparti les données annotées en deux ensembles : 2/3 pour l'entraînement (i.e. créer les modèles de classification) et la validation (i.e. sélectionner les meilleurs paramètres pour ces modèles) et 1/3 pour le test (i.e. effectuer l'évaluation finale) [TRE]. Parmi tous les concepts, une dizaine ont été retenus pour l'évaluation : Building, Car, Explosion/Fire, US Flag, Maps, Mountain, Prisoner, Sports, People walking/running, Waterscape sur 43776 plans pour TRECVid'05. Ces concepts apparaissent de 31 à 2549 fois comme le montre le tableau 1.3.

Chaque participant soumet un certain nombre de résultats (runs), qui consiste en une liste ordonnée par concept des 2000 premiers plans vidéos identifiés par le système d'indexation. La vérité-terrain a été jugée par la méthode de "pooling", ou l'ensemble des documents retournés ne seront pas tous vérifiés (i.e. seulement 75000 plans évalués sur 45000 x 10

[5] Nous avons fait le choix de conserver la terminologie technique en anglais dans un souci de cohérence du manuscrit, afin d'éviter toute confusion, en particulier dans le champ des noms de concepts et des descripteurs MPEG-7 qui seront introduits dans le prochain chapitre.

[6] Nous avons participé à un effort commun d'annotation fourni par l'ensemble des participants à l'aide de l'outil de [AQ07] pour l'évaluation TRECVid 2007.

Id TREC	Concepts	Pos.Devel	Pos.Test	Pos.Total
1	Sports	600	330	930
9	Building	1311	1238	2549
12	Mountain	138	56	194
17	Waterscape	315	98	413
25	Prisoner	31	0	31
28	US Flag	185	75	260
30	Car	1028	396	1424
34	Walking/Running	1732	439	2171
36	Explosion/Fire	281	22	303
38	Maps	405	253	658

Tab. 1.3 – Id des concepts TRECVid 2005.

concepts de l'ensemble de test). Ainsi, le pourcentage de plans vidéos jugés positifs varie entre 0.8% et 45.8%, ce qui implique que l'évaluation 2005 est moins fiables pour les participants qui expérimentent des algorithmes originaux, comme il est décrit dans la vue d'ensemble de la campagne 2005 exposée dans [OIKS05].

Enfin, en 2006, le corpus a doublé en conservant toutes les données de 2005 comme ensemble d'entraînement, pour identifier cette fois-ci 39 concepts parmi lesquels 20 ont été évalués : Airplane, Boat/Ship, Bus, Car, Court, Desert, Gouvernement, Natural Disaster, Office, Outdoor, Person, Police Security, Prisoner, Road, Sky, Sports, Urban, Vegetation, Waterscape, Weather.

1.4.1.2 TRECVid 2007

La campagne 2007 a été conduite par une nouvelle collection de 100 heures de vidéos complètement différentes des deux précédentes. Elle propose d'analyser le comportement de nos systèmes d'indexation face à une large sélection de vidéos couleurs et en noire et blanc, en introduisant des documentaires, reportages, émissions scientifiques, programmes éducatif et vidéos d'archives. Quelques 50 heures de vidéos sont utilisées pour la tache d'entraînement et de validation pour détecter une liste de 36 concepts (Table 1.5) et 50 heures pour l'évaluation de nos systèmes à travers le protocole commun d'évaluation de TRECVid basé sur la métrique précision et le rappel, établis par la "précision moyenne". Ce point sera détaillé dans la partie évaluation 3.3.

Enfin, il est intéressant de relever que notre tache dans TRECVid présente certains points caractéristiques qu'il faut prendre en considération lors de la conception de notre système d'indexation :

1. **La Grande dimension** de la base de donnée (e.g. 100H de vidéos pour TRECVid 2007), les descripteurs qui représentent les plans vidéos sont souvent de grande dimension. Il est important de traiter ce problème si l'on ne veut pas souffrir des effets de la *malédiction de la dimension* (traité dans la section 3.4.2).

2. **Classes complexes :** Les 39 concepts sémantiques LSCOM-lite [7] [NKK+05] présentent un certain niveau de difficultés du moment ou plusieurs d'entre eux sont très proches (comment dissocier entre la classe PERSON et PRISONER, ou OUTDOOR et URBAN ?). Celles-ci peuvent être étudiées dans le cadre de la similarité inter-concepts et de l'ontologie (traité dans le chapitre 4).

3. **Déséquilibre pertinent/non pertinent :** La taille de la catégorie recherchée est généralement très petite devant la taille de la base. Il y a 1000 fois plus d'éléments non-pertinents que de pertinents par concept (traité dans la section 3.2.1.4).

4. **Satisfaction de l'utilisateur :** La qualité d'un système de recherche du contenu sémantique dans les plans vidéos est avant tout jugée par ses utilisateurs. Afin de modéliser cette satisfaction, TRECVid utilise un critère particulier, comme la précision moyenne que nous présenterons dans la partie 3.3.

5. **Rapidité :** Le but est de concevoir un système qui peut être utilisé rapidement et efficacement. Ainsi, on préférera une technique très peu coûteuse en calculs : il est difficile de concevoir l'attente de l'utilisateur pendant plusieurs minutes entre chaque interaction. En ce sens, nous nous restreindrons à des techniques de faible complexité, sauf dans les cas particuliers d'étude.

1.4.2 Vidéos de match de football

Pour l'heure, les supporters peuvent se connecter sur l'opérateur téléphonique pour suivre la dernière actualité, les exploits et de vivre les grands moments de leurs équipes. Un système de navigation propose aux abonnés, le suivi des scores en temps réel, ainsi que l'accès aux vidéos, aux résumés des matchs, de découvrir la vidéo du but dans les minutes qui suivent l'action, etc. Le dispositif *Orange-foot* se distingue par une approche très grand public. Ainsi, 8 matchs de chaque journée de championnat sont diffusés en direct sur le mobile. Des magazines de 5 à 10 minutes font les résumés de chaque journée et des meilleures actions. Les abonnés pourront également bénéficier de toute l'information footballistique en direct : scores, résumés, classements, feuilles de matchs, calendrier.

Dans la continuité, nous avons travaillé sur les données du projet CRE-Fusion, en utilisant des vidéos de sport tirées de matchs de football. Dans un premier temps, nous avons effectué l'annotation des images-clés de chaque plan dans les deux mi-temps du match France/Portugal (Table 1.4) ainsi que la vérité-terrain. L'annotation consistait à décrire les plans vidéos et les images-clés par des mots choisis dans un vocabulaire prédéfini par Orange-France Télécom Labs. Au total, le vocabulaire comptabilise 11 concepts ou classes. L'unité de traitement et d'évaluation est l'image-clé. Nous avons appliqué l'outil proposé par Volkmer et al. [VTW04] pour la segmentation en plans. En effet, les 96 minutes de vidéos (905 plans de 3385 images-clés) ont été annotées et regroupées en deux ensembles : 2/3 pour l'entraînement et 1/3 pour l'évaluation.

[7] LSCOM : Large Scale Concept Ontology for Multimedia. URL *http ://www.lscom.org*

Id	Concepts	Pos.Devel	Pos.Test	Pos.Total
1	Close-Up Action	617	200	817
2	Game Stop	76	81	157
3	Lateral Camera	92	50	142
4	Goal Camera	13	5	18
5	Global Center View	507	217	724
6	Global Rear View	6	13	19
7	Global Right View	142	21	163
8	Global Left View	208	144	352
9	Zoom on Public	94	139	233
10	Zoom on Player	246	156	402
11	Aerial View	15	15	30

Tab. 1.4 – Id des concepts. La quantité relative de chaque classe est précisée pour donner une idée de la borne inférieure des performances à obtenir.

1.5 Conclusion

Dans ce chapitre, nous avons introduit les systèmes de recherche d'images et de vidéos par le contenu. Nous avons décrit leurs structures générales et les difficultés qui peuvent être rencontrées pour mettre en place avec précision le cadre des travaux présentés dans les prochains chapitres par rapport à un contexte national et international. Enfin, une brève présentation du support vidéo a été effectuée, en particulier sur la segmentation temporelle en scènes, en plans et la sélection des images-clés. Ainsi, le plan sera l'unité de base pour le projet NoE K-Space, et l'image-clé pour CRE-Fusion.

Notre prochaine étude se concentrera sur les caractéristiques visuelles qui peuvent être extraites du plan vidéo. Ensuite, nous présenterons les méthodes de classification utilisées dans cette thèse.

Par ailleurs, une question intéressante peut être posée concernant la capacité de l'être humain à détecter le contenu ou les objets dans une image. Quelles sont les caractéristiques qui lui permettent d'identifier les concepts dans des images visuellement différentes, comme le montre la Fig. 1.8 ? Les réponses à ce genre de questions seront introduites dans le prochain chapitre, inspirées des techniques d'apprentissage.

Fig. 1.8 – Exemple de difficultés qui peuvent être résolues par l'apprentissage. Grâce à l'apprentissage, l'être humain arrive facilement à reconnaître la "pomme" qu'elle soit complète ou déjà entamée, malgré que les trois images présentent des formes et des caractéristiques visuellement différentes.

Id	Id TREC	Concepts	Pos.Devel	Pos.Test	Pos.Total
1	1	Sports	106	42	148
2	3	Weather	51	34	85
3	4	Court	113	5	118
4	5	Office	921	453	1374
5	6	Meeting	548	270	818
6	7	Studio	358	468	826
7	8	Outdoor	3437	1812	5249
8	9	Building	1116	477	1593
9	10	Desert	61	15	76
10	11	Vegetation	1465	499	1964
11	12	Mountain	76	17	93
12	13	Road	660	297	957
13	14	Sky	1303	853	2156
14	15	Snow	36	91	127
15	16	Urban	1334	537	1871
16	17	Waterscape	355	414	769
17	18	Crowd	921	552	1473
18	19	Face	5484	2325	7809
19	20	Person	7099	2972	10071
20	23	Police Security	256	63	319
21	24	Military	232	74	306
22	25	Prisoner	13	7	20
23	26	Animal	405	271	676
24	27	Computer Tv	463	202	665
25	28	US Flag	10	0	10
26	29	Airplane	28	7	35
27	30	Car	417	187	604
28	31	Bus	47	40	87
29	32	Truck	95	19	114
30	33	Boat/Ship	101	151	252
31	34	Walking/Running	859	385	1244
32	35	People Marching	120	82	202
33	36	Explosion/Fire	12	19	31
34	37	Natural Disaster	19	21	40
35	38	Maps	50	31	81
36	39	Charts	126	80	206

Tab. 1.5 – Id des concepts TRECVid 2007.

Chapitre 2

Description et Classification des Caractéristiques Visuelles

Cette partie présente un état de l'art des technologies de description du contenu audiovisuel, basées sur les descripteurs bas-niveau, dont ceux considérés par le standard MPEG-7. Dans un premier temps, nous allons définir ce qu'est un bon descripteur bas-niveau. Puis, un état de l'art sur la description visuelle image et vidéo est exposé avec la méthodologie employée pour l'extraction. Enfin, nous porterons une attention particulière à la classification, où trois méthodes seront présentées.

2.1 Description bas-niveau

2.1.1 Qu'est ce qu'un bon descripteur bas-niveau ?

Un bon descripteur est celui qui décrit le contenu avec une grande variance, pour être capable de distinguer tout type de média, en prenant en compte la complexité de l'extraction, de la taille du descripteur codé, de l'échelle et de l'interopérabilité [1]. MPEG-7 définit un ensemble de descripteurs, où quelques uns ne sont que des structures d'agrégation ou de localisation de certains descripteurs [Eid03]. Dans les paragraphes suivants, nous allons mettre en avant les plus importants descripteurs utilisés dans cette thèse.

2.1.2 Descripteurs visuels MPEG-7

MPEG-7 est un standard ISO/IEC 15398 développé par le *Moving Picture Experts Group (MPEG)* en 2001. Cette norme a été élaborée par des centres de recherche universitaires et des industriels. A la différence avec MPEG-1, MPEG-2 et MPEG-4, qui standardisent l'encodage même des documents audiovisuels, rendant respectivement possible l'apparition de la vidéo interactive sur le CDROM (VCD, SVCD, DVD), la télévision digitale (satellite, câble ou encore ADSL), le téléchargement, le *streaming* sur Internet, le multimédia sur

[1] L'interopérabilité est la capacité que possède un système intégralement connue, à fonctionner avec d'autres systèmes existants ou futurs.

mobile et la télévision haute définition, etc. Quant au MPEG-7 appelé aussi *"Multimedia Content Description Interface"*, régit la description des documents audiovisuels et non l'encodage. La navigation, la recherche et le filtrage peuvent donc s'opérer sur ces descriptions plutôt que sur les fichiers sources, ce qui évite de devoir décompresser ces derniers et de les traiter à chaque requête.

2.1.2.1 Descripteurs de couleur

La couleur est le descripteur le plus utilisé dans la recherche d'images et de vidéos par le contenu. Elle a été très étudiée durant les deux dernières décennies. La version actuelle de MPEG-7 [Mpe01a, Mpe01b] inclut un certain nombre d'histogrammes qui sont capables de capturer la distribution de la couleur avec une précision raisonnable pour la recherche et la détection d'image. Cela dit, il y a quelques dimensions à prendre en considération, incluant le choix de l'espace de couleur, de la quantification de l'espace de couleur et des valeurs de l'histogramme. MPEG-7 supporte 5 espaces de couleurs : RGB, HSV, YCrCb, HMMD et Monochrome. L'espace HMMD est composé de Hue (H), Maximum (Max), Minimum (Min) des valeurs RGB et la Différence entre les valeurs Max et Min. Les descripteurs sont définis de préférence dans les espaces non-linéaires (HSV, HMMD) proche de la perception humaine de la couleur. Maintenant, nous allons présenter les détails techniques de chaque descripteur :

1. **Dominant Color Descriptor (DCD)** spécifie un ensemble de couleurs dominantes dans l'image, où un petit nombre de couleurs est suffisant pour caractériser l'information couleur [Cie00]. Dans [IRB02], l'auteur démontre qu'en moyenne, on a besoin entre 4 et 6 couleurs pour créer un modèle d'histogramme de couleurs. Le DCD est obtenu par la quantification des valeurs des pixels dans un ensemble de N couleurs dominantes c_i, présentées comme les composantes du vecteur suivant.

$$DCD = \{(c_i, p_i, v_i), s\} \quad i = \{1, 2, ..., N\} \tag{2.1}$$

   où p_i est le pourcentage de pixels dans l'image et v_i la variance associée à c_i. s est la cohérence spatiale (i.e. le nombre moyen de pixels en connections avec une couleur dominante, utilisant une fenêtre masque 3x3).

2. **Color Layout Descriptor (CLD)** est une représentation compacte de la distribution spatiale des couleurs [KY01]. Il est obtenu en utilisant la combinaison entre la structure en bloc et le descripteur de couleurs dominantes, comme suit (Fig. 2.1) :
   - *Partitionnement :* l'image est divisée en (8x8) blocs pour garantir l'invariance de la résolution ou de l'échelle [MOVY01] ;
   - *Sélection des couleurs dominantes :* pour chaque bloc, une couleur dominante est sélectionnée. Les blocs sont transformés en une série de coefficients en utilisant la couleur dominante ou la couleur moyenne, pour obtenir les composantes $CLD = \{Y, Cr, Cb\}$, où Y est le coefficient de luminance, Cr et Cb pour la chrominance ;
   - *Transformation DCT :* les trois composantes (Y, Cb et Cr) sont transformées en trois ensembles de coefficients DCT ;

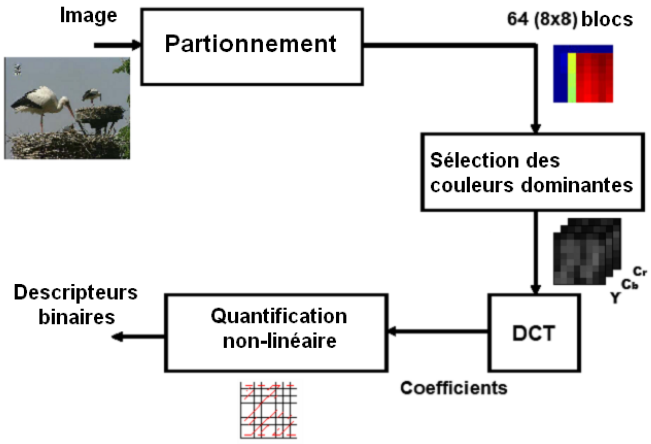


FIG. 2.1 – Etapes d'extraction du descripteur CLD.

- *Quantification non-linéaire :* les coefficients basse fréquence sont extraits en utilisant le balayage en zigzag et quantifiés, formant le CLD d'une image fixe.

3. **Color Structure Descriptor (CSD)** code la structure locale de la couleur dans une image en utilisant un élément structurant de dimension (8x8) comme le montre la Fig. 2.2. Le CSD est obtenu en visitant tous les pixels de l'image. Ensuite, la fréquence d'occurrence des couleurs dans chaque élément structurant est représentée par quatre possibilités de quantification dans l'espace de couleur HMMD : 256, 128, 64 et 32 bins [MBE01].

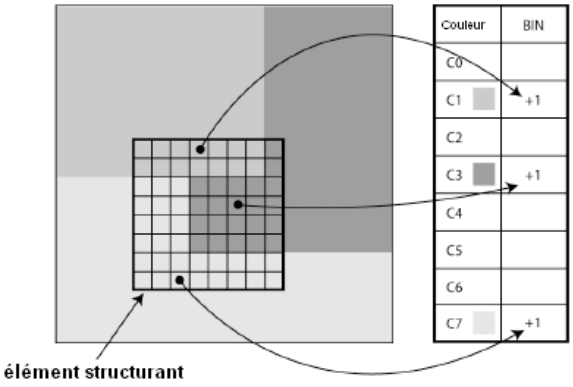


FIG. 2.2 – Représentation de l'élément structurant pour le calcul du descripteur CSD [MBE01].

4. **Scalable Color Descriptor (SCD)** est définit comme une quantification uniforme de l'espace de couleur hue-saturation-value (HSV). Les valeurs des bins subissent une quantification non-linéaire pour réaliser un encodage efficace. La différence réside dans l'utilisation de la transformation de Haar (filtre de Haar) par le SCD, s'exprimant ainsi dans le domaine fréquentiel. Cette transformation fournit une description compacte et une représentation multi-échelles de l'histogramme [Mpe01a]. L'intérêt de cette multi-résolutions est le passage du grossier au raffinement. Ce descripteur ne tient pas compte de la structure locale de la couleur. Le codage par la transformation de Haar est utilisée pour réduire le nombre de bins dans l'histogramme original à 16, 32, 64, 128 où 256 bins.

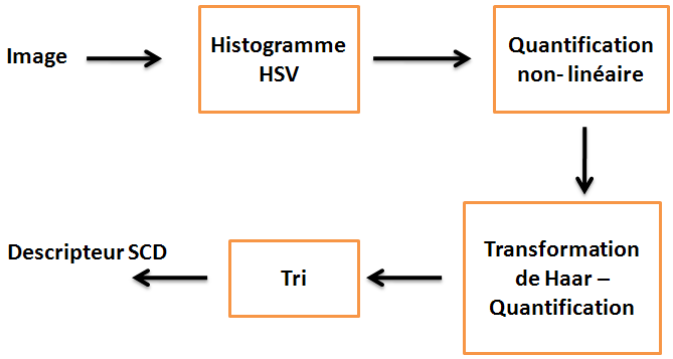


FIG. 2.3 – Etapes d'extraction du descripteur SCD.

5. **Group of Frame/Picture Descriptor (GoF/GoP)** sont obtenus en combinant le SCD de chaque frame/image comprise dans le groupe [MSS02]. Trois GoF/GoP descripteurs sont proposés à base de la moyenne, médiane et d'intersection des histogrammes SCD.

2.1.2.2 Descripteurs de Texture

La texture, comme la couleur, est un descripteur bas-niveau très efficace pour l'indexation et la recherche par le contenu. Le standard MPEG-7 propose trois descripteurs de texture : Homogeneous Texture, Browsing Texture, Edge Histogram (seulement deux seront traités dans ce mémoire).

1. **Homogeneous Texture Descriptor (HTD)** caractérise la texture régionale utilisant des statistiques de fréquences locales. La HTD est extraite par le filtre de Gabor. Les filtres de Gabor ont la particularité de faire un filtrage proche de celui réaliser par notre perception visuelle. Ils sont sensibles à l'orientation et à la fréquence [MOVY01]. Un ensemble de filtres permet alors de capturer les directions et les fréquences principales de l'image. Un filtre de Gabor est un filtre de fréquence pure modulé par une gaussienne. Trois paramètres doivent être spécifiés par l'utilisateur, la fréquence maximale, le nombre d'orientations et le nombre d'échelles. Les travaux de Manjunath et al. [MM96, RKK+01] présentent 6 échelles, 5 canaux d'orientation et la fréquence

maximale de 0.5, comme le meilleur choix de paramètre de direction et d'angle, donnant au total 30 canaux, comme le montre la Fig. 2.4. Ensuite, l'énergie e et sa déviation d sont calculées pour chaque canal [MSS02, XZ06].

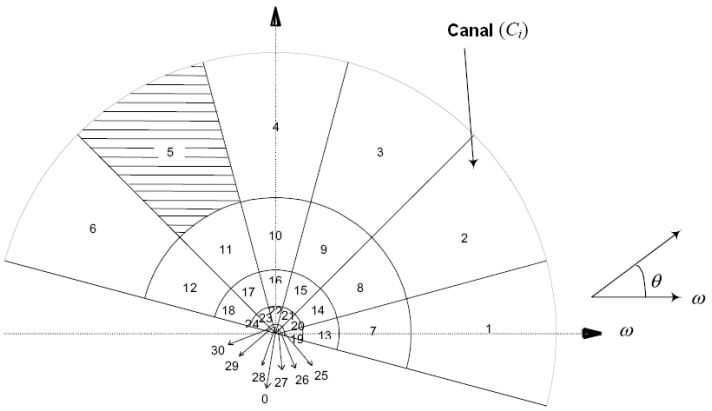


FIG. 2.4 – Exemple de partitionnement de l'espace de fréquence pour le descripteur HTD (6 temps de fréquence, 5 canaux d'orientation).

Après filtrage, le premier et le second moments (*avg*, std) dans les 30 canaux de fréquences sont calculés, composant le descripteur HTD de 62-dimensions (Eq. 2.2).

$$HTD = [avg, \text{std}, e_1, ..., e_{30}, d_1, ..., d_{30}] \tag{2.2}$$

2. **Edge Histogram Descriptor (EHD)** exprime la distribution locale des contours dans l'image. Un histogramme de contour représente la fréquence et l'orientation des changements de luminosité dans l'image, essentiellement sur 5 types de bords (0°, 45°, 90°, 135°, en plus du non-orientation). Plus précisément, l'image est divisée en (4x4) sous-images sans chevauchement. Pour chaque sous-image, nous générons un histogramme de contours de 80-dimensions (16 sous-images, 5 types de bords). Le descripteur peut être améliorer en ajoutant un niveau global et semi-global sur la localisation des contours dans l'image [PJW00].

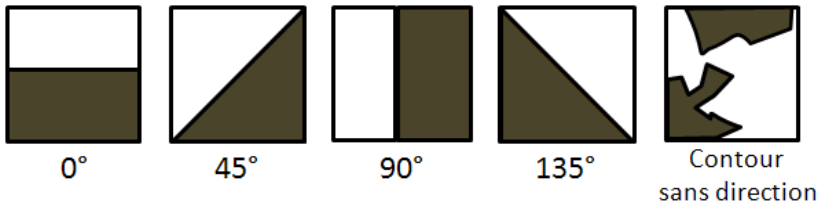


FIG. 2.5 – Types de contours.

– L'histogramme global des contours résume la distribution des contours sur toute l'image, il est obtenu par l'addition des histogrammes locaux de chaque type de bords pour obtenir un vecteur de 5 bins.

– L'histogramme semi-global des contours est obtenu par l'addition des histogrammes locaux de 13 segments différents, comme le montre la Fig. 2.6, générant un vecteur de 65-dimensions (13 sous-images, 5 types de bords).

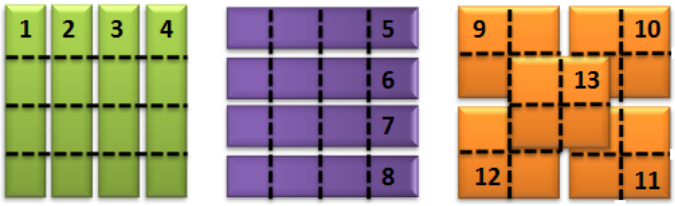


FIG. 2.6 – Segments de sous-images pour l'histogramme semi-global.

2.1.2.3 Descripteurs de Formes

Semblable à la couleur et à la texture, la forme présente un rôle important dans la détection d'objets. Les descripteurs de formes sont moins développés que les deux précédents descripteurs due à la complexité de la représentation des formes. Dans le standard MPEG-7, trois descripteurs sont proposés : contour-based shape, region-based shape et 3D-shape. Dans cette thèse, nous avons utilisé la première qui sera détaillée ci-dessous.

1. **Contour-based Shape Descriptor (C-SD)** présente un objet 2D fermé ou un contour de région dans l'image. Il caractérise les propriétés du contour. Dans [ZL03], il a été conclu que le descripteur à base de *Curvature Scale Space* (CSS) offre la meilleure performance globale, et est robuste aux changements d'échelle, à la translation et à la rotation.

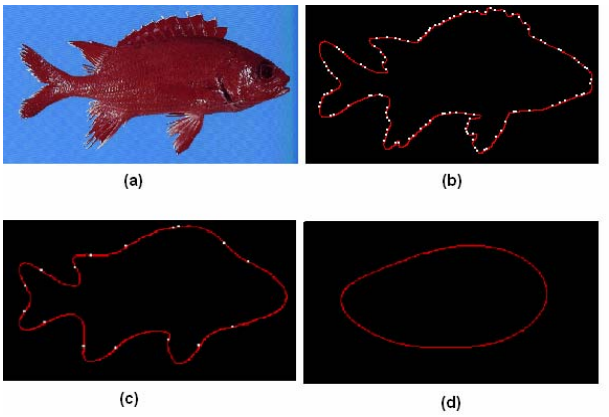


FIG. 2.7 – Représentation de CSS pour un contour de poisson. (a) image originale, (b) N points initialisés sur le contour, (c) après t itérations (filtrage passe bas), (d) contour final convexe (Figure tirée de la présentation de M. Vajihollahi [VF02])

Le CSS consiste à suivre les positions des points d'inflexion d'un contour, alors que celui-ci subit une série de filtrage répétitif passe bas sur les coordonnées des points x et y sélectionnés du contour (Fig. 2.7). Au fur et à mesure des itérations de filtrage, le contour devient de plus en plus lisse et les inflexions non significatives sont éliminées. Les points d'inflexion qui subsistent sont considérés comme étant caractéristiques du contour. Ensuite, les points de séparation concaves, les parties convexes du contour et les pics (maxima du contour de la carte CSS) sont identifiés, et les valeurs normalisées sont enregistrées dans le descripteur. Des caractéristiques supplémentaires telles que l'*excentricité* [2] , la *circularité*, le *nombre de sommets* du contour original et filtré peuvent être introduites dans la formation du descripteur [MSS02].

2.1.2.4 Descripteurs de Mouvement

Les descripteurs de mouvements peuvent être très utiles pour la détection du contenu dans les séquences vidéos, introduisant sa dimension temporelle, qui est généralement très coûteuse en termes de temps de calcul et du grand volume d'informations. On présentera quatre descripteurs caractérisant les différents aspects du mouvement : mouvement de la camera, activité, trajectoire et le mouvement paramétrique.

1. **Camera Motion Descriptor (CMD)** détaille le mouvement fourni par une caméra à un instant donné dans une scène. Le CMD prend en compte le positionnement physique de la caméra, particulièrement l'axe optique tels que le panoramique horizontal *panning* ou le panoramique vertical *tilting*, que les mouvements engendrés par une modification de la focale comme pour le *zoom* avant ou arrière. Tous ces changements au niveau de la caméra induisent un mouvement global dans un plan image [MSS02].

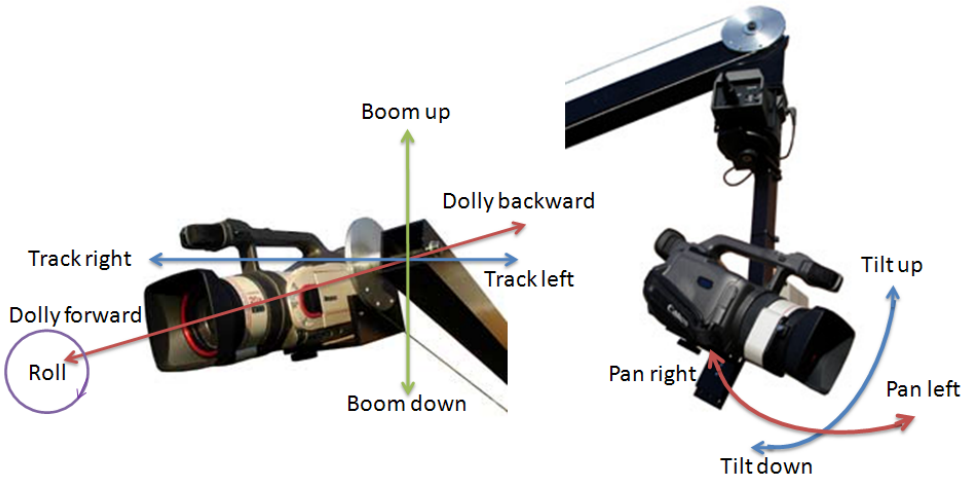


FIG. 2.8 – Mouvements de base d'une caméra. Les plus étudiés sont *"pan, tilt, zoom"*.

[2] L'excentricité est un paramètre caractéristique d'une courbe conique. En fonction de sa valeur, on obtient soit un cercle ($e = 0$), ellipse ($e =]0, 1[$), parabole ($e = 1$), hyperbole ($e > 1$).

2. **Motion Activity Descriptor (MAD)** indique comment une scène peut être perçue par un téléspectateur comme : lente, rapide ou action se déroulant à un certain rythme [SMD02]. Un exemple d'une *haute activité* est l'action de but dans un match de football ou de tennis, tandis qu'une *faible activité* comprend les journaux télévisées, les scènes d'interviews, etc. Des attributs complémentaires peuvent également être extraits. On citera :
   - *L'intensité du mouvement :* basée sur les déviations standards des amplitudes du vecteur mouvement. Les déviations standards sont quantifiées en cinq valeurs d'activités, une valeur élevée indique une forte activité et une petite valeur d'intensité indique une faible activité.
   - *La direction du mouvement* exprime la direction dominante quand le plan vidéo présente plusieurs objets avec des activités différentes.
   - *La distribution spatiale de l'activité* indique la taille et le nombre de régions actives dans l'image. Par exemple, le visage de l'interviewer aurait une seule grande région active, tandis que la séquence de rue aurait plusieurs petites régions actives.
   - *La distribution temporelle de l'activité* exprime la variation de l'activité sur la durée du plan vidéo.

3. **Motion Trajectory Descriptor (MTD)** présente le déplacement des objets au cour du temps. Le MTD est défini comme une localisation spatio-temporelle donnée par la position d'un point représentatif comme le centre de masse. La trajectoire est représentée par une liste de $N-1$ vecteurs, où N est le nombre de frames dans une séquence vidéo [MSS02]. Cependant, quand la vidéo est longue, la description de chaque frame demande un temps de calcul et un espace de stockage élevé. Pour cela, un échantillonnage des frames peut réduire les effets de deux limites précédemment citées [YLK+01]. Le schéma de description proposé représente chaque trajectoire de

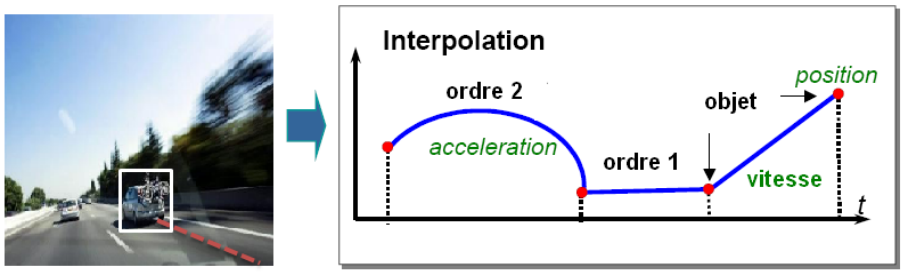


FIG. 2.9 – Représentation de la trajectoire du mouvement (1 dimension).

mouvement d'un objet, en utilisant une fonction polynomiale (Eq. 2.3). Les paramètres i, x_i, y_i, v_i, et a dénotent l'index du segment, la position initiale du centroïde (x, y), la vitesse de l'objet et son accélération dans chaque segment. Le modèle de trajectoire approxime la position spatiale du centroïde au premier ou au second ordre (voir la

Fig. 2.9).

$$\begin{cases} x(t) = x_i + v_{x_i}(t - t_i) + \frac{1}{2}a_{x_i}(t - t_i)^2 \\ y(t) = y_i + v_{y_i}(t - t_i) + \frac{1}{2}a_{y_i}(t - t_i)^2 \end{cases} \tag{2.3}$$

4. **Parametric Motion Descriptor (PMD)** représente le mouvement global des objets dans une séquence vidéo par un des modèles paramétriques du tableau 2.1 [MSS02].

TAB. 2.1 – Modèles paramétriques du mouvement.

Modèles	Équation
Translation (2 paramètres)	$\begin{cases} v_x(x,y) = a_1 \\ v_y(x,y) = a_2 \end{cases}$
Rotation/échelle (4 paramètres)	$\begin{cases} v_x(x,y) = a_1 + a_3x + a_4y \\ v_y(x,y) = a_2 - a_4x + a_3y \end{cases}$
Affine (6 paramètres)	$\begin{cases} v_x(x,y) = a_1 + a_3x + a_4y \\ v_y(x,y) = a_2 + a_5x + a_6y \end{cases}$
Perspective (8 paramètres)	$\begin{cases} v_x(x,y) = (a_1 + a_3x + a_4y)/(1 + a_7x + a_8y) \\ v_y(x,y) = (a_2 + a_5x + a_6y)/(1 + a_7x + a_8y) \end{cases}$
Quadratique (12 paramètres)	$\begin{cases} v_x(x,y) = a_1 + a_3x + a_4y + a_7xy + a_9x^2 + a_{10}y^2 \\ v_y(x,y) = a_2 + a_5x + a_6y + a_8xy + a_{11}x^2 + a_{12}y^2 \end{cases}$

où $v_x(x,y)$ et $v_y(x,y)$ représentent le déplacement des composantes du pixel de coordonnée (x,y). Le PMD spécifie le modèle du mouvement, l'intervalle de temps, les coordonnées et la valeur du paramètre a_i.

Pour estimer le mouvement global dans une image, la méthode d'optimisation régulière peut être utilisée, où les variables sont les paramètres du modèle choisi et l'erreur de compensation du mouvement est la fonction à minimiser (Eq. 2.4) entre l'image du mouvement transformé $I^{'}$ et l'image référence I.

$$\zeta = \sum_i (I^{'}(x_i^{'}, y_i^{'}) - I(x_i, y_i))^2 \tag{2.4}$$

2.1.3 Propriétés des descripteurs visuels MPEG-7

Les descripteurs de couleurs produisent des histogrammes avec de faible nombre de bins, favorisant une rapide indexation et recherche des requêtes. Toutefois, les descripteurs CLD et SCD présentent un processus d'extraction plus complexe que les descripteurs DCD et SCD, ce qui augmente le temps de calcul. Par ailleurs, les descripteurs de texture HTD (62 bins) et EDH (80 ou 150 bins) sont faciles à extraire, mais impliquent un temps considérable

de navigation. Concernant le descripteur de formes C-SD, il est en général simple à extraire et très discriminant. Il est invariant à l'orientation et à l'échelle [MSS02]. Cependant, deux inconvénients sont à mentionner : (1) le C-SD ne permet pas de saisir les caractéristiques globales, ce qui amène à une pauvre précision de la méthode CSS pour les courbes qui ont un petit nombre de concavités et de convexités. (2) Ils sont exigeants en temps de calcul lors de l'extraction. D'autre part, le mouvement produit une description cohérente, utile et complémentaire aux descripteurs précédents. On notera que la taille du MTD est proportionnelle au nombre de points et des coordonnées du centre de masse. Enfin, l'utilisation du standard MPEG-7 est le résultat d'une concertation interne entre les différents partenaires du projet NoE K-Space, afin de pouvoir travailler la même base de description des plans vidéos, tout en permettant l'introduction d'autres informations.

2.1.4 Autres descripteurs

Dans cette partie, nous allons décrire les descripteurs extraits à partir des régions dans l'image. Pour cela, une étape de segmentation sera nécessaire.

2.1.4.1 Segmentation

La segmentation d'image a pour but de regrouper des pixels entre eux suivant des critères pré-définis, constituant une partition de l'image. Ce problème particulièrement difficile qui n'a actuellement pas de solution générique est largement traité dans la littérature. L'étude et l'analyse de nouveaux algorithmes de segmentation spatiale n'entre pas dans le cadre de nos travaux de thèse. Nous allons donc simplement présenter la méthode pour laquelle nous avons opté afin de réaliser cette tâche.

Les images-clés des plans vidéos sont segmentées en régions homogènes en utilisant deux méthodes de segmentation : (1) en bloc, (2) segmentation de graphe [FH98]. Cette dernière a été sélectionnée pour sa capacité à préserver les détails dans les images avec peu de variations et de les ignorer dans les images avec de grandes variations. De plus, l'algorithme est assez rapide, permettant une segmentation en $O(n \log n)$ pour un graphe à n arêtes. Une illustration des résultats des deux méthodes est présentée par la Fig. 2.13.

Segmentation en blocs

La première méthode de segmentation utilisée dans nos travaux, en particulier sur les données du projet CRE-Fusion, est celle d'une division de l'image (576x720 pixels) en un certain nombre de (6x6) sous-images sans chevauchement, comme le montre la Fig. 2.10. On obtient 36 blocs représentatifs de 96x120 pixels chacun. La couleur, la texture et les contours sont ensuite extraits, plus de détails sur l'extraction des caractéristiques sont donnés dans le partie 2.1.4.2. Notons que le nombre de blocs est choisi en prenant en compte deux points : (1) l'obtention de blocs de taille représentatif (i.e. possédant assez de pixels pour une bonne représentation de la couleur, les contours, etc) et (2) se prévenir d'une grande dimension qui risque d'être produite dans l'étape d'extraction des descripteurs.

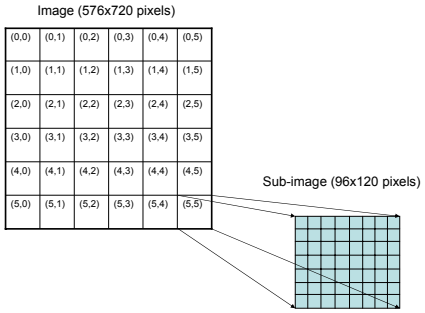


FIG. 2.10 – Segmentation en blocs d'une image.

Segmentation en régions à base de graphe

Ici, nous allons brièvement décrire l'algorithme de segmentation de graphe décrit dans Felzenszwalb et al. [FH98] et qui sera utilisé dans nos travaux. L'algorithme cherche à prouver l'existence de limites (frontières) entre chaque paire des régions voisines. Initialement, les noeuds du graphe non orientés $G = \{V, E\}$ sont identifiés aux pixels de l'image (les éléments de V) et les arcs sont construits entre un pixel et ses quatre voisins (en haut, en bas, à gauche et à droite). Chaque arc e est pondéré par un poids $w(e)$ mesurant la différence entre les deux pixels connectés par cet arc (e.g. la différence d'intensité, de couleur, de texture, etc).

Dans cette représentation, la segmentation S revient à faire un partitionnement de V en région, où chaque région $C_{i=\{1\cdots k\}}$ est un arbre recouvrant le poids minimum (ou *Minimum Spanning Tree (MST)*) et définit ainsi un ensemble de pixels connexes sur l'image. Ainsi, une suppression de tous les bords entre les différents pixels x tel que $w(x_i, x_j) > \tau$. Chaque région est donc représentée par sa variation interne $Int(C_i)$ (i.e. par le plus grand poids $w(e)$ de C_i). Deux régions se comparent par leur variation externe $Ext(C_i, C_j)$, correspondant au poids minimum des arcs reliant C_i et C_j comme le montre l'équation (Eq. 2.5).

$$\begin{cases} Int(C_i) = \max_{e \in \mathrm{MST(C,E)}} w(e) \\ Ext(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)) \end{cases} \tag{2.5}$$

La partition voulue est obtenue sur la base de l'algorithme de Kruskal [3] [Pri57]. Au départ $S = \{C_1, C_2, \ldots, C_{|V|}\}$, aucun sommet n'est connecté. On ajoute ensuite les arcs par ordre croissant de leur poids, en évitant les cycles comme le montre la Fig. 2.11. Un critère d'arrêt $D(C_i, C_j)$ permet de limiter la croissance des régions :

$$D(C_i, C_j) = \begin{cases} 1 & \text{si} \quad Ext(C_i, C_j) \succ \min(Int(C_i) + \tau_i, Int(C_j) + \tau_j) \\ 0 & \text{ailleurs} \end{cases} \tag{2.6}$$

[3] Il est possible d'utiliser aussi l'algorithme de Prim [Kru56].

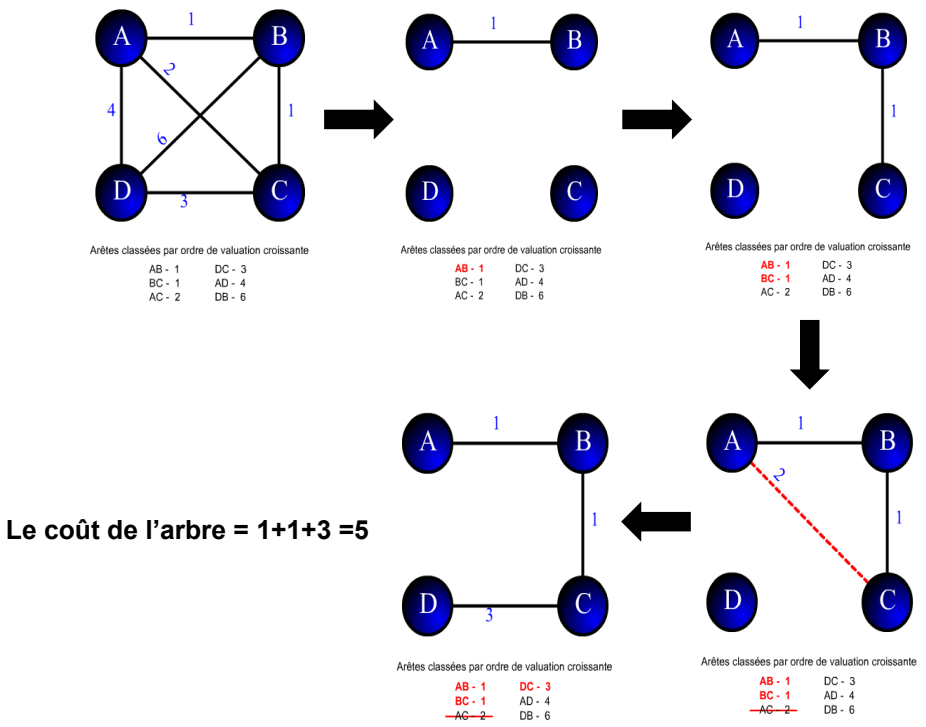
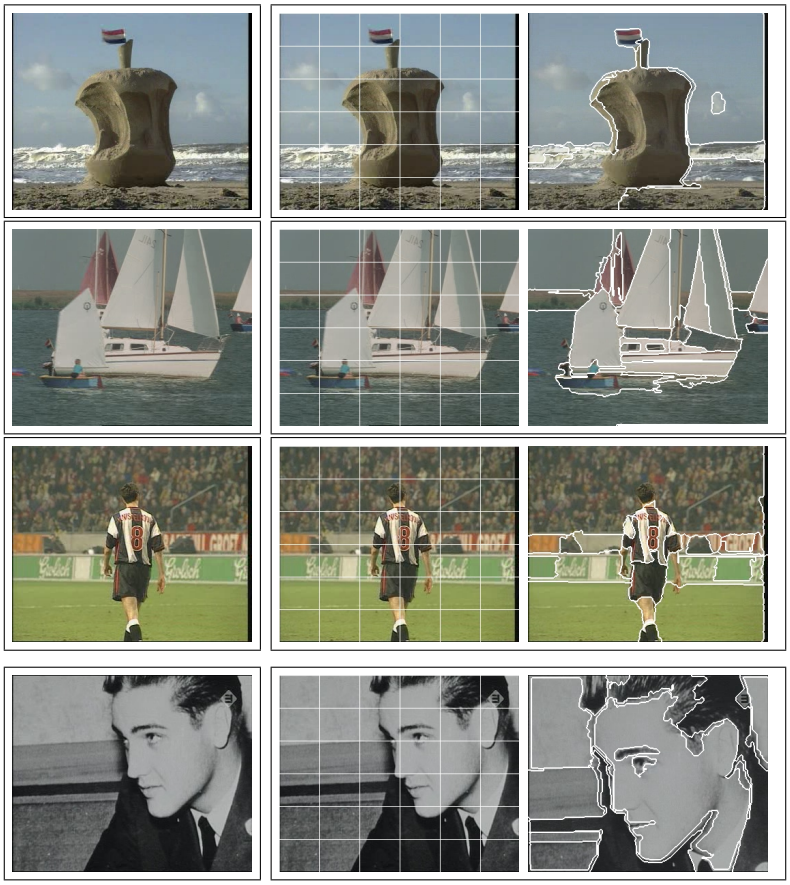


FIG. 2.11 – Exemple de fonctionnement de l'algorithme de Kruskal sur un graphe à 4 noeuds.

FIG. 2.12 – Résultats de la segmentation obtenue avec trois valeurs du paramètre K (100, 200 et 300) qui module la finesse de la segmentation.

Le seuil τ sert à pénaliser le regroupement des grandes régions, il diminue avec la taille de la région. Selon [FH98], nous définissons $\tau_i = K/|C_i|$, où $K > 0$ est une constante et $|C_i|$ est le nombre de pixels dans la région C_i. Plus K augmente, il favorise un regroupement de régions comme le montre la Fig. 2.12. La propriété principale de l'algorithme est de respecter un critère de bonne segmentation à partir de D. La segmentation est dite :

- *trop fine* si $D = 0$ entre deux régions adjacentes (i.e. elles doivent être regroupées) ;
- *trop grossière* si $D = 1$, il est possible de partitionner une des régions.



(a) référence (b) Segmentation

FIG. 2.13 – Exemples de résultats de la segmentation sur des images couleurs et monochrome. Source : collection de TRECVid.

2.1.4.2 Description des régions

La segmentation étant réalisée, la prochaine étape est la description des régions. Nous avons retenu deux types de caractéristiques :

- **La couleur :** nous avons opté pour l'utilisation de deux espaces de couleur RGB et HSV. Ce dernier est une transformation non-linéaire de l'espace RGB, et est perceptiblement plus approprié.
- **La Texture :** Deux formes de texture sont représentées :

  1. **Par les filtres de Gabor :** la texture est modélisée par les énergies des réponses de 24 ou 48 filtres de Gabor décrites dans la partie de *HomogeneousTexture*. Nous avons choisi les paramètres habituels, c'est à dire une fréquence maximale de 0.5, six orientations et quatre échelles [BH06].

  2. **Par les contours :** les contours d'une image sont considérés comme des caractéristiques importantes pour représenter le contenu vu la sensibilité de la perception

humaine. Pour localiser la distribution des contours, on utilise la méthode de segmentation de l'image en blocs décrite dans la partie 2.1.2.2. Ainsi, pour chaque bloc, on génère un histogramme de contours. Cinq directions (0°, 45°, 90°, 135°, Non-direction) seront détectées [PJW00].

2.1.5 Construction du dictionnaire visuel et de la signature

Après avoir effectué l'extraction des informations visuelles (i.e. couleurs, textures, etc) de chaque région des images-clés sous forme d'histogramme. Cette représentation appelée parfois signature ou index, est nécessaire pour résumer les descripteurs en une forme plus exploitable par le système. Pour cela, un dictionnaire visuel est calculé en utilisant l'algorithme de regroupement "K-means". Cet algorithme a été sélectionné pour la simplicité de sa mise en oeuvre et sa rapidité d'exécution, permettant de déterminer un nombre prédéfini de centres qui représentent au mieux l'ensemble des histogrammes. Ces centres (appelés aussi *mots-clés visuels*) composent alors un dictionnaire visuel par caractéristique. Ainsi, nous allons construire autant de dictionnaires que de caractéristiques extraites.

Ensuite, chaque région sera associée à un ou plusieurs mots-clés visuels du dictionnaire visuel en utilisant une mesure de distance ; Euclidienne dans notre cas. Par ailleurs, on peut dire que la signature est simplement le dénombrement des mots-clés visuels. Elle permet la comparaison entre deux signatures, en étudiant leurs mots-clés visuels communs. Une mesure numérique de la similarité de deux signatures peut être obtenue par le calcule du produit scalaire ou le cosinus. La Fig.2.14 illustre le principe qui vient d'être décrit.

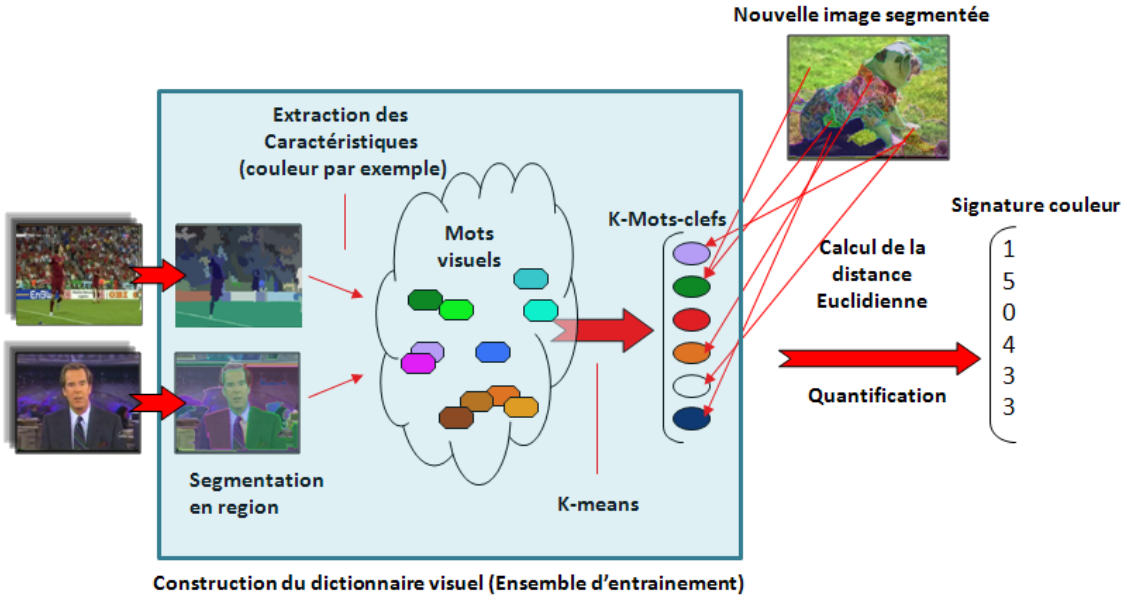


FIG. 2.14 – Etapes de construction d'une signature.

Dans Souvannavong et al. [Sou05], une étude comparant les résultats de la classification

en fonction du nombre de centres choisis pour la quantification sur les données de TRECVid 2005, montre qu'à partir de 1000 centres, les performances ne sont plus diminuées par la quantification. Le nombre de centres à sélectionner est actuellement un problème non résolu. Seuls des essais et l'observation des performances moyennes permettent de le choisir, sachant qu'il est étroitement lié aux requêtes (e.g. dans [Sou05], où les expériences ne sont pas reprises ici, on observe que pour le concept FILLE, il faut prendre 2000 centres et 1000 pour le concept REQUIN). Par ailleurs, les auteurs relèvent que la signature résultante souffre de :

- la segmentation, qui est rarement robuste aux changements de luminosité, de contraste ou au mouvement, ce qui produit un décalage de l'information résultante (i.e. les éléments d'une scène ne sont pas segmentés de la même manière).
- l'imprécision résultant de la quantification (i.e. deux régions visuellement proches peuvent être décrite par deux mots-clés visuels différents).

Pour résoudre ces deux problèmes, ils proposent d'appliquer la méthode d'analyse de la sémantique latente (LSA) initialement utiliser par Deerwester et al. [DDL+90] dans le but d'établir des relations de synonymie et polysémie entre les mots en indexation des documents textuels. Cela revient à chercher les mots sémantiquement similaires dans un texte. Ainsi, l'utilisation de la SVD "décomposition en valeurs singulières" de la matrice d'occurrence proposée par Souvannavong et al. [Sou05], permettra de mettre en valeur les relations existantes entre les mots, en représentant la matrice d'occurrences C de taille (NxM) sous la forme :

$$C = UDV^t \tag{2.7}$$

avec, $UU^t = U^tU = I_N, V^tV = I_N$ et $D = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_m)$.

La décomposition SVD en soi n'apporte aucun changement. Par contre, des effets très intéressants sont observés en approximant la matrice diagonale D par ses p plus grandes valeurs singulières $\{\lambda_i\}$. La matrice obtenue $\hat{C}_p$ contient de nouveaux coefficients indiquant de nouvelles relations entre les mots-clés visuels et les plans.

De la même manière que pour le choix des centres, seuls les expérimentations peuvent donner une idée sur le nombre de valeurs propres à sélectionner. On note que si le nombre de valeurs singulières supprimées est élevé, il y aura plus d'équivalences créées y compris celles qui n'auraient pas lieu d'être. Si toutes les valeurs singulières sont conservées, aucune relation n'est établie et les signatures ont facilement tendance à être différentes à cause des problèmes précédemment cités. Les résultats présentent une augmentation de 10% de la précision moyenne en conservant seulement entre 6% et 8% des composantes des 2 000 centres. Les signatures de couleur et de texture sont donc réduites à une taille de moins de 200 composantes pour 2 000 mots-clés visuels, réduisant la taille et le temps de calcul.

2.1.6 Discussion

Au début de cette thèse, nous avons repris et utilisé les travaux de Souvannavong et al. [Sou05] sur la LSA, particulièrement sur les données TRECVid 2005, qui rappelons le présente des plans vidéos de journaux télévisées avec un certain nombre réduit de concept.

Or, les effets attendus de la LSA n'ont pas été visibles sur les données de TRECVid 2007 qui introduisent des vidéos noir et blanc d'archive, des dessins animés et des vidéos complètement différentes de celles de TRECVid 2005. Par conséquent, nous avons décidé de ne pas l'appliquer pour la suite de nos expérimentations et de travailler plus particulièrement sur la conception d'un système générique basé sur la fusion multi-niveaux par rapport au processus de classification. Pour cette raison, la prochaine section sera dédiée à la présentation des méthodes de classification utilisées dans nos travaux. Cependant, nous avons gardé la même structure pour la construction du dictionnaire visuel et de la signature.

2.2 La Classification

La classification est un processus qui permet l'estimation de la ou les classes auxquelles un élément/ objet x donné appartient. Elle peut être menée de deux manières : aveugle ou supervisée. Les méthodes aveugles (non supervisée) permettent de déterminer automatiquement des classes au sein d'un ensemble puis de classer les éléments dans ces classes. On citera comme exemple l'algorithme *K-means*. Contrairement aux méthodes supervisées qui nécessitent un ensemble d'entraînement [4]. L'entraînement permet alors de calculer les paramètres des modèles sélectionnés afin de minimiser les erreurs de classification. Le modèle entraîné permet ensuite de classer de nouveaux éléments inconnus. On citera comme exemple le *réseau de neurones,* le *mélange de gaussiennes,* etc.

Cependant, la qualité des modèles dépend de leur capacité de généralisation [5]. Toutefois, cette estimation est délicate et souvent un ensemble de validation est utilisé pour valider le modèle [Sou05].

Nos travaux de thèse se place dans le cadre d'une classification supervisée. Par ailleurs, dans [ASS00], les auteurs montrent qu'il est possible de transformer un problème de classification multi-classes en plusieurs problèmes bi-classes en utilisant le principe du *un-contre-tous* (« one-against-all »). Chaque système binaire classifie les échantillons dans une classe ou dans une autre qui comprend toutes les classes restantes. Cette méthode est adoptée dans notre système de classification.

Dans ce qui va suivre, nous allons exposé les algorithmes les plus couramment utilisés en classification supervisée par la communauté, particulièrement : mélanges de gaussiennes, réseau de neurones et les machines à vecteurs de support.

2.2.1 Modèle de Mélange de Gaussienne (GMM)

Si on a des événements issus d'une composition de N phénomènes de distributions inconnus. Dans ce cas, nous pouvons les modélisés avec des distributions que l'on connaît, de paramètres que l'on optimise pour les faire «coller» au mieux. De ce fait, la densité

[4] Un ensemble d'entraînement est constitué de couples formés par le descripteur d'un élément et les classes associées.

[5] La généralisation peut se définir comme la capacité de classer correctement de nouveaux éléments qui sont différents de ceux présents dans l'ensemble d'entraînement. Une bonne généralisation permet de garantir une estimation correcte lorsque les échantillons présentés diffèrent de ceux présents dans l'apprentissage.

prendra la forme d'une composition de lois Normales [6].

Un mélange de gaussiennes m_G est une fonction de densité de probabilité définie par la somme pondérée de plusieurs densités de probabilité qui suivent une loi gaussienne. Elle permet de prendre en compte la diversité des classes et la présence de sous-classes par l'intermédiaire des mélanges et de la variabilité.

$$\begin{aligned} m_G(x) &= \sum_{i=1}^{n_G} \alpha_i \mathcal{N}(\mu_i, \Sigma_i)(x) \\ \text{avec } \mathcal{N}(\mu_i, \Sigma_i)(x) &= \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \\ \text{et} \sum_{i=1}^{n_G} \alpha_i &= 1 \end{aligned}$$

Les paramètres $\alpha_i, \theta_i(\mu_i, \Sigma_i)$ sont respectivement le poids de la i^{me} gaussienne, sa moyenne et sa matrice de covariance, avec n_G fixé à priori. La principale difficulté de cette approche consiste à déterminer le meilleur paramètre (α, θ). Pour cela, on cherche habituellement le paramètre qui maximise la vraisemblance pour η individus :

$$L(x, \alpha, \theta) = \sum_{k=1}^{\eta} log\left(m_G(x_k)\right) \tag{2.8}$$

Ces paramètres sont souvent estimés en utilisant l'algorithme itératif EM, qui repose sur une optimisation itérative des paramètres du modèle (moyennes, matrices de covariances, et probabilités à priori des composantes du mélange) [PJKKK95]. Par ailleurs, les étapes E et M peuvent être conjointement réalisables comme décrit ci-dessous :

$$p(c_j/x_i, \theta_i^t) = \frac{\alpha_j \mathcal{N}(x_i/\theta_j^t)}{\sum_{v=1}^{n_G} \alpha_j \mathcal{N}(x_i/\theta_v^t)} \tag{2.9}$$

$$\mu_j^{t+1} = \frac{\sum_{i=1}^{\eta} p(c_j/x_i, \theta_i^t)\ x_i}{\sum_{i=1}^{\eta} p(c_j/x_i, \theta_i^t)} \tag{2.10}$$

$$\Sigma_j^{t+1} = \frac{\sum_{i=1}^{\eta} (x_i - \mu_j^{t+1})(x_i - \mu_j^{t+1})^T p(c_j/x_i, \theta_i^t)}{\sum_{i=1}^{\eta} p(c_j/x_i, \theta_i^t)} \tag{2.11}$$

$$P(c_j/x_i, \theta_i^{t+1}) = \alpha_j = \frac{1}{\eta} \sum_{i=1}^{\eta} p(c_j/x_i, \theta_i^t) \tag{2.12}$$

Une fois l'estimation effectuée, chaque individu se verra attribuer la classe à laquelle il appartient le plus probablement. Cependant, en ce qui concerne le nombre de gaussienne à estimer n_G, plusieurs critères peuvent être utiliser pour aider à un choix automatique, on

[6] l'idée des GMM se base sur le théorème de la limite centrale : "La moyenne de M variables aléatoires lorsque $M \longrightarrow \infty$ tend vers une gaussienne". Donc, on peut espérer qu'un phénomène naturel qui a plusieurs causes aléatoires en concurrence à une distribution globale qui tend à être gaussienne

citera les critères statistiques comme BIC (Bayesian Information Criteria) et ICL (Integrated Completed Likelihood), etc [Aya07] ou des critères propres au problème tel que la précision moyenne.

2.2.2 Réseau de Neurones (NN)

Un réseau de neurones est un système composé de plusieurs unités (ou neurones) de calcul simples fonctionnant en parallèle, connectées entre elles par des liaisons affectées de poids. Ces liaisons permettent à chaque neurone de disposer d'un canal pour envoyer et recevoir des signaux en provenance d'autres neurones du réseau. Chacune de ces connexions reçoit un poids (une pondération), qui détermine sont impact sur les neurones qu'elle connecte. Chaque neurone dispose ainsi d'une entrée, qui lui permet de recevoir de l'information des autres neurones, mais aussi de ce que l'on appelle une fonction d'activation [7], qui est dans les cas les plus simple, une simple identité du résultat obtenu par l'entrée et enfin une sortie.

L'exemple le plus utilisé d'un réseau de neurones est le perceptron multi-couches. Dans un perceptron, plusieurs couches contenant des neurones sont connectées entre elles de l'entrée vers la sortie. Chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Afin d'illustrer un peu ces propos, la Fig. 2.15 représente le schéma type d'un perceptron à trois couches :

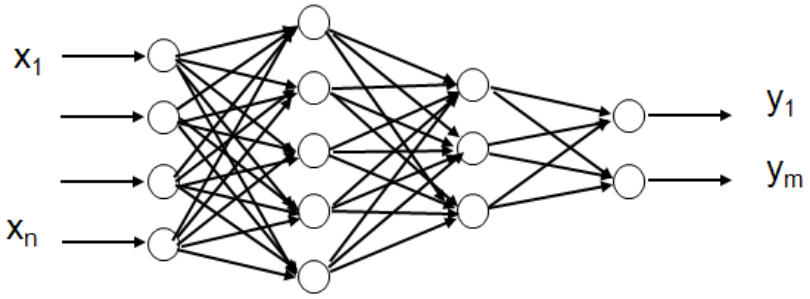


Fig. 2.15 – Réseau de neurones multi-couches à deux couches cachées.

- **La couche d'entrée** reçoit les données source que l'on veut utiliser pour l'analyse. Dans notre cas, cette couche recevra les descripteurs bas-niveau des plans vidéos. Sa taille est donc directement déterminée par le nombre de variables d'entrées.
- **Les couches cachées** n'ont qu'une utilité intrinsèque pour le réseau de neurones, sans contact direct avec l'extérieur. Les fonctions d'activations sont souvent non linéaires sur cette couche mais il n'y a pas de règle à respecter. Le choix de sa taille n'est pas implicite et doit être ajusté. En général, on peut commencer par une taille moyenne des couches d'entrée et de sortie mais ce n'est pas toujours le meilleur choix. Il sera souvent préférable pour obtenir de bon résultats, d'essayer le plus de tailles possibles.

[7] Le choix de la fonction d'activation est un élément important pour le réseau de neurones. Comme, l'identité n'est pas toujours suffisante. Souvent, des fonctions non linéaires et plus évoluées seront nécessaires comme : la fonction logistique : $f(x) = \frac{1}{1+\exp(-d.x))}$, la tangente hyperbolique : $f(x) = \frac{2}{1+\exp(-2x)} - 1$, la fonction gaussienne : $f(x) = \exp(-\frac{X^2}{2})$ et la fonction à seuil : $f(x) = 0$ si $X < 0$ et $f(x) = 1$ si $X > 0$

– **La couche de sortie** donne le résultat obtenu après compilation par le réseau des données d'entrée. Dans notre cas, cette couche donne les concepts détecter dans le plan vidéo. Sa taille est directement déterminée par le nombre de variables qu'on veut en sortie.

L'apprentissage consiste tout d'abord à calculer les pondérations optimales des différentes liaisons, en utilisant un échantillon de donnée. La méthode la plus utilisée est la *rétro-propagation* : on introduit des valeurs de la couche d'entrée et en fonction de l'erreur obtenue en sortie (le delta), on corrige les poids accordés aux pondérations. C'est un cycle qui est répété jusqu'à ce que la courbe d'erreurs du réseau ne soit croissante (il faut bien prendre garde de ne pas sur-entraîner un réseau de neurones qui deviendra alors moins performant) ou après un certain nombre d'itérations préalablement fixé.

Il existe plusieurs type de réseau de neurones dans la littérature, les chercheurs n'ont de cesse que d'inventer de nouveaux types de réseaux toujours mieux adaptés à la recherche de solutions de problèmes particuliers. Notre intérêt s'est porté sur un cas particulier des MLP, qui est le réseau RBF (Radial Basis Function) [Bis95]. L'avantage principal des RBF est qu'il est possible de fortement simplifier l'apprentissage en divisant le travail en trois, comme expliqué dans la suite. Pour autant que le nombre de données soit suffisant, les résultats obtenus avec ce modèle sont aussi bons que ceux du MLP.

Dans un premier temps, on va brièvement décrire le réseau RBF (Fig. 2.16) qui est constitué des couches suivantes :

– *La couche d'entrée :* propage l'information donnée par l'entrée ;
– *La couche RBF :* couche cachée qui contient les neurones RBF [8]. Les neurones sont des gaussiennes centrées sur un point de l'espace d'entrée : $f(x) = \exp(\frac{x^2}{2\sigma^2})$.
– *La couche de sortie :* couche simple qui contient une fonction linéaire. Ainsi, la sortie du réseau est une combinaison linéaire des sorties des neurones RBF multipliés par le poids de leurs connexions respectives.

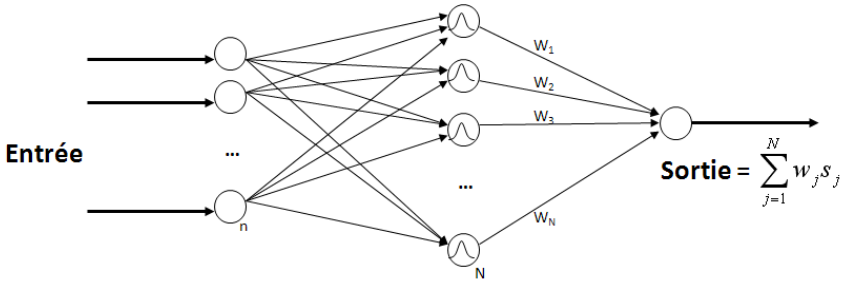


FIG. 2.16 – Structure du réseau RBF.

L'apprentissage d'un RBF se déroule en trois phases : positionnement des centres (en relation avec le nombre de neurones choisi dans l'unique couche cachée), détermination de la largeur des noyaux gaussiens et adaptation des poids. Toute modification d'un de ces

[8] Théoriquement, toute transformation non-linéaire pourrait être utilisée telle que la sigmoïde. En pratique, les fonctions radiales donnent de bons résultats pour les problèmes d'approximation. Notre attention s'est tournée vers la fonction radiale de forme gaussienne.

paramètres entraîne directement un changement du comportement du réseau.

1. Le nombre de neurones RBF (N) et la position des gaussiennes sont deux paramètres intimement liés. Deux optiques sont possibles selon le nombre d'élément I dans la base d'apprentissage, soit :
   - I n'est pas trop grand et alors $N = I$. Dans ce cas-ci (le plus simple), le nombre de neurones RBF est égale au nombre d'exemples soumis au réseau. Chacune des gaussiennes est alors centrée sur un des exemples ;
   - I est trop grand et on choisi $N << I$. Dans ce cas-ci, le nombre de neurones RBF devient un véritable paramètre. Il n'existe pas de méthode pour le déterminer. Il s'agit donc de trouver le nombre de centroïdes adéquat lié au problème donné. Une fois choisi, il faut déterminer leur position via une quantification vectorielle (Learning Vector Quantization LVQ) par exemple. Cette solution permettra d'obtenir la meilleure répartition des centroïdes possible.
2. Une fois tous les centres c_j choisis, il faut déterminer la largeur σ des gaussiennes. Deux règles empiriques peuvent être prise :
   - si on choisit un σ égal pour toutes les gaussiennes. $\sigma = \frac{d}{\sqrt{M}}$, avec M = nombre de centroïdes et $d = max||c_i - c_j||$, $1 \leq i, j \leq M$;
   - du moment ou rien ne nous impose de prendre le même σ pour chaque centroïde, on pourra utiliser $\sigma_i = \frac{1}{M\sqrt{8}} \sum_{i=1}^{N} ||x_i - c_j||$.
3. Le nombre et la position des centroïdes et la largeur des gaussiennes fixées, les poids w de chacune des connexions (RBF-output) peuvent être calculés par l'équation matricielle suivante :

$$\begin{pmatrix} f(||x_1 - c_1||) & \cdots & f(||x_1 - c_M||) \\ f(||x_2 - c_1||) & \cdots & f(||x_2 - c_M||) \\ \vdots & \ddots & \vdots \\ f(||x_N - c_1||) & \cdots & f(||x_N - c_M||) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \cdots \\ w_M \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{pmatrix}$$

2.2.3 Machines à Vecteur de Support (SVM)

SVM [Vap00] est l'algorithme de classification le plus populaire de nos jours, permettant d'étendre la notion du neurone à des configurations plus complexes des données. Son principe repose sur la recherche d'une séparatrice dans un espace de grande dimension lorsque les données d'entrées ne sont pas linéairement séparables dans l'espace d'origine. Elle présente deux forces : (1) la maximisation des marges (i.e. grandeur mesurant l'écart du modèle aux points de l'entraînement) autour de l'hyperplan séparateur lui assure de bonnes capacités de généralisation et (2) la représentation des données par un noyau [9] lui permet de résoudre des problèmes non-linéairement séparables.

[9] Une fonction noyau satisfaisant les conditions de Mercer peut être exprimée par un produit scalaire dans un espace de dimension supérieur. Les conditions de Mercer garantissent qu'il existe un espace $\mathcal{F}$ et une fonction $\Phi : \mathcal{E} \rightarrow \mathcal{F}$ tel que $K(x, y) = \Phi(x).\Phi(y)$

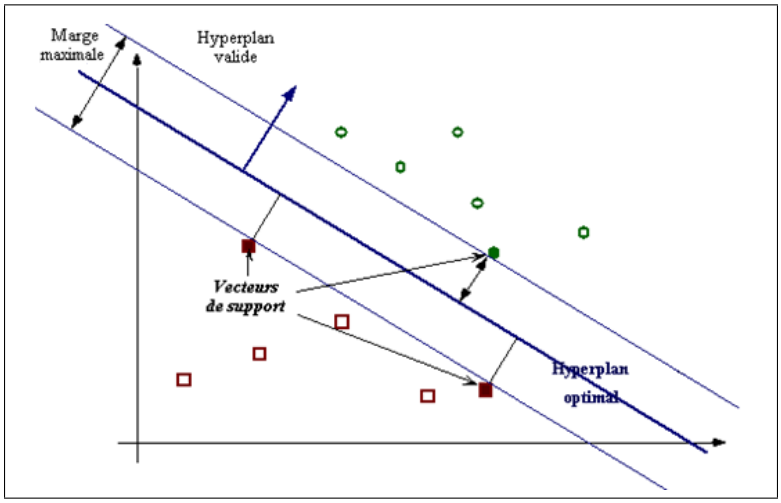


FIG. 2.17 – Séparation linéaire dans un espace à deux dimensions.

Plus formellement, soit $X = \{x_i\}$ l'ensemble des données étiquetées suivant $c = \{-1, 1\}$, qui représente la classe de chaque individu. L'idée de base est la recherche d'un hyperplan $H(x) = sign(wx + b)$ paramétré par (w, b) qui sépare deux classes linéairement :

$$H(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{sinon} \end{cases} \tag{2.13}$$

La frontière de décision $H(x) = 0$ est un hyperplan séparateur. Rappelons que le but est d'apprendre $H(x)$ par le biais d'un ensemble d'apprentissage $N_{\mathcal{L}}$.

$$c_i(w.x_i + b) \geq 1 \tag{2.14}$$

Le choix du séparateur n'est pas évident. Il existe en effet une infinité de cas, dont les performances en apprentissage sont identiques, mais les performances en généralisation peuvent être très différentes. Il a été montré qu'il existe un unique hyperplan optimal, défini comme l'hyperplan qui maximise la marge avec les échantillons [Vap00]. La distance d'un point x à l'hyperplan est égale à $H(x)/||w||$

Afin de maximiser la marge $\Delta = \frac{2}{||w||}$ (Fig. 2.17), il est nécessaire de minimiser $\|w\|$. L'utilisation des multiplicateurs de Lagrange permettent de montrer que :

$$w_{opt} = \sum_{i=1}^{N_{\mathcal{L}}} \alpha_i c_i x_i \tag{2.15}$$

$$\sum_{i=1}^{N_{\mathcal{L}}} \alpha_i c_i = 0 \tag{2.16}$$

$$\alpha_i \geq 0 \tag{2.17}$$

L'ensemble des paramètres de Lagrange $\alpha_i \neq 0$ définit les vecteurs de support v_i qui sont utilisés. L'hypothèse construite est :

$$\mathcal{H}_{SVM}(x) = b_{opt} + \sum_{i=1}^{N_v} \alpha_i v_i x \tag{2.18}$$

Cette étude peut être généralisée au cas non linéaire en remplaçant le produit scalaire par une fonction noyau $K(x, y)$ qui respecte les conditions de Mercer [CT00]. Les noyaux les plus communément utilisés actuellement sont les noyaux polynomiaux $K(x, y) =< F(x), F(y) >^d$ et gaussiens $K(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$.

Dans cette thèse, nous avons utilisés la librairie SVMLight (Joachims [5]) avec un noyau gaussien [10] pour réaliser nos expériences. Ce noyau possède un paramètre de lissage σ qui doit soit être prédéfini ou fixé par validation croisée. L'ensemble de validation est alors utilisé pour estimer au mieux les trois paramètres intervenant dans le modèle (i.e. le lissage, le nombre de composantes et le coût des erreurs). Nous obtenons en sortie des distances d_i qui serons transformer en une estimation d'appartenance aux classes, à travers l'adoptions de la fonction sigmoïde pour le calcul du degré de confidence y_i^j.

$$y_i^j = \frac{1}{1 + \exp\left(-\beta d_i\right)} \tag{2.19}$$

où (i, j) représente le $i^{\text{ème}}$ concept et $j^{\text{ème}}$ descripteur bas-niveau. β est un paramètre obtenu expérimentalement qui représente la pente.

2.3 Conclusion

Ce chapitre a été dédié à la représentation du contenu dans les plans vidéo. Un état de l'art sur les méthodes d'extraction des caractéristiques visuelles a été présenté avec la description de la méthodologie employée. Ensuite, nous avons abordé le thème de la classification en présentant trois des méthodes utilisées dans cette thèse.

Le prochain chapitre va détailler comment peut ont combiner cet ensemble de caractéristiques hétérogènes de manière efficace, en tenant compte des multiples difficultés exposées précédemment, le chapitre propose plusieurs contributions, concernant à la fois la fusion des descripteurs bas-niveau (*feature fusion*) et haut-niveau (*classifier fusion*) dans une optique d'indexation automatique des plans et images-clés par le contenu.

[10] Le noyau gaussien a été préféré après que l'utilisation de ce type de noyau dans un cas similaire de classification [Sou05], a donné de bonnes performances comparant à d'autres choix.

Chapitre 3

Fusion Multi-niveaux pour l'Analyse Multimédia

Aujourd'hui, le nombre d'études théoriques qui traitent de la classification et de l'apprentissage automatique d'un point de vue général, sans s'attacher à une application particulière, reste très faible. Bien qu'elles abordent les vrais problèmes comme celui de la combinaison, elles utilisent des hypothèses rarement vérifiées. Cela ne permet pas d'avoir une approche générale qui traite le problème de la combinaison de manière satisfaisante dans tous les cas de figure. La mise en oeuvre d'une telle approche n'ira pas sans poser certaines questions : Comment modéliser la combinaison des informations produites par des descripteurs bas-niveau ou des classifieurs et d'exploiter leurs complémentarités pour un problème donné ? En fonction de quels critères et comment l'évaluer ? Comment évaluer la robustesse d'une méthode donnée sur des applications différentes ? Ces questions restent des problèmes ouverts et constituent un défi à relever, ainsi, nous allons répondre à ces interrogations.

Ce chapitre est décomposé en deux grandes parties. La première traite de la fusion de descripteurs haut-niveau (fusion tardive) et la deuxième de la fusion de descripteurs bas-niveau (fusion précoce). Dans un premier temps, un état de l'art est conduit et nous en profitons pour justifier nos choix et les propositions apportées, en particulier l'adaptation de la théorie des évidences au réseau de neurones, donnant ainsi le "Neural Network based on Evidence Theory (NNET)". Cette théorie a l'avantage de présenter deux nouvelles informations importantes pour la prise de décision comparée aux méthodes probabilistes : l'ignorance du système et le degré de croyance. Ensuite, le NNET a été amélioré en intégrant les relations entre descripteurs et concepts, modélisées par un vecteur de pondération basé sur l'entropie et la perplexité. La combinaison de ce vecteur avec les sorties de classifieurs, nous donne le "Perplexity-based Evidential Neural Network (PENN)". Dans un second temps, nous aborderons la fusion bas-niveau. Ceci n'a été possible qu'après une étude statistique des données avant et après la fusion des caractéristiques. Enfin, l'évaluation des résultats sur deux types de bases de données est exposée, dans chacun des deux niveaux de combinaison. Avant cela, nous allons introduire de façon générale cette notion de "fusion".

3.1 La Fusion

Ces dernières années, la fusion d'informations est un domaine qui connaît une évolution importante, en particulier dans l'indexation et la recherche des documents multimédia où les sources d'informations se sont multipliées, qu'il s'agisse de capteurs, d'informations à priori, des caractéristiques (image, audio, texte et mouvement), etc. Chaque source d'information étant en général imparfaite et insuffisante, il est important d'en combiner plusieurs afin d'avoir une meilleure connaissance du «monde». La fusion d'informations peut alors se définir comme :

La combinaison d'informations souvent imparfaites et hétérogènes, afin d'obtenir une information globale plus complète, de meilleure qualité, permettant de mieux décider et d'agir.

On pourra aussi citer quelques définitions, qui ont globalement, toutes à peu près la même signification [Ram01].

1. **Roger Reynaud** *La fusion de données décrit les méthodes et les techniques numériques permettant de mélanger des informations provenant de sources différentes (nous parlerons aussi de modalités différentes) afin d'obtenir une décision ou une estimation.*
2. **Mongi A. Abidi** *Data fusion deals with the synergetic combination of information made available by various knowledge sources such as sensors, in order to provide a better understanding of a given scene.*
3. **M. Kokar** *Multisensor fusion is defined as the process of combining inputs from sensors with information from other sensors, information processing blocks, databases, or knowledge bases, into one representational format.*

Une des caractéristiques importantes de l'information est son imperfection. Elle peut prendre les formes suivantes :

- **l'incertitude** est relative à la vérité d'une information et caractérise son degré de conformité à la réalité ;
- **l'imprécision** concerne le contenu de l'information et mesure donc son défaut quantitatif de connaissance ;
- **l'incomplétude** caractérise l'absence d'information apportée par la source sur certains aspects du problème ;
- **l'ambiguïté** exprime la capacité d'une information de conduire à deux interprétations. Elle peut provenir des imperfections précédentes ;
- **le conflit** caractérise deux ou plusieurs informations conduisant à des interprétations contradictoires et donc incompatibles. Les situations conflictuelles sont fréquentes dans les problèmes de fusion et posent toujours des problèmes difficiles à résoudre.

D'autres caractéristiques de l'information peuvent être vues comme positives et sont exploitées pour limiter les imperfections.

- **la redondance** apporte plusieurs fois la même information, elle est observée dans la mesure où les sources donnent des informations sur le même phénomène, cela permet de réduire les incertitudes et les imprécisions ;

– **la complémentarité** apporte des informations sur des grandeurs différentes du phénomène observé. Elle est exploitée directement dans le processus de fusion pour avoir une information globale plus complète et pour lever les ambiguïtés.

3.2 Fusion de descripteurs haut-niveau

La fusion de descripteurs haut-niveau (*High-Level Feature HLF*) appelée aussi fusion/ combinaison de classifieurs, a été proposée comme une voie de recherche permettant de fiabiliser la reconnaissance en utilisant la complémentarité qui peut exister entre les classifieurs. Sur ce point, la littérature abonde de travaux présentant des méthodes de combinaison qui se différencient aussi bien par le type d'informations apportées par chaque classifieur que par leurs capacités d'apprentissage et d'adaptation. Cette section a pour objectif de présenter les recherches actuelles sur la combinaison parallèle de classifieurs. Tout d'abord, nous rappelons la problématique générale de la combinaison parallèle et proposons une dichotomie des méthodes de combinaison en fonction de critères que nous justifions. Nous détaillons les méthodes les plus connues dans la littérature ainsi que les développements récents dans le domaine. Ensuite, nous présenterons une nouvelle méthode de combinaison basée sur la théorie des évidences NNET. Enfin, une version améliorée qu'on appellera PENN "Perplexity-based Evidential Neural Network" sera proposée.

3.2.1 État de l'art

La combinaison parallèle de classifieurs peut se présenter de la façon suivante : étant donné un ensemble de L classifieurs se prononçant chacun indépendamment sur une même forme à reconnaître, comment élaborer une réponse finale unique à partir des L résultats fournis (Fig. 3.1). Ce problème de combinaison parallèle nécessite d'abord de rappeler ce qu'on entend généralement par « classifieur » dans le cadre de la combinaison puis d'examiner les critères à prendre en compte pour catégoriser les différentes méthodes de combinaison présentées dans la littérature.

3.2.1.1 Définition d'un classifieur dans le cadre de la combinaison

Nous appelons *classifieur* tout outil de reconnaissance qui reçoit un élément x (e.g. forme, objet, etc) en entrée et donne des informations à propos de la classe de celle-ci. Cet outil est vu comme une fonction qui, à l'aide des descripteurs de x à reconnaître, décide de lui attribuer la classe C_i parmi un nombre fini de classes possibles $i = \{1, ..., M\}$.

Les réponses fournies par un classifieur peuvent être divisées en deux catégories suivant le type d'information apporté :

– type classe : $e_j(x) = C_i(i \in \{1, ..., M\})$, indique que le classifieur j a attribué la classe C_i à x.
– type mesure : $e_j(x) = (m_1^j, m_2^j, ..., m_M^j)$ où m_i^j est la mesure attribuée à la classe i par le classifieur j.

Chaque type de sortie correspond à un niveau d'information différent fourni par le classifieur. La sortie de type classe est la plus générale mais apporte le moins d'informations. La

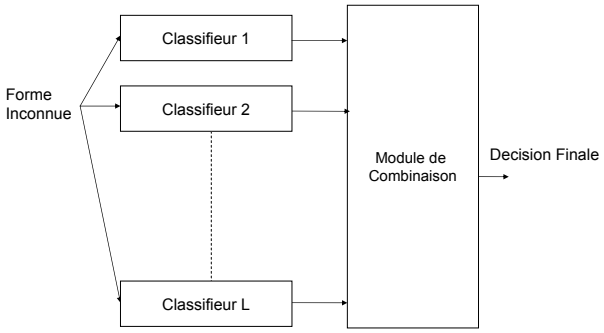


FIG. 3.1 – Schéma de combinaison parallèle de classifieurs.

sortie de type mesure est plus riche en information puisqu'elle reflète le niveau de confiance du classifieur dans ses propositions. Toutefois, ces mesures ne sont pas toujours comparables (une distance, une probabilité à *posteriori*, une valeur de confiance ou une fonction de croyance), par conséquent, une normalisation est souvent nécessaire.

3.2.1.2 Dichotomie des méthodes de combinaison

Plusieurs regroupements des méthodes de combinaison ont été proposées dans la littérature [KBD01, DT00, XKS92]. Dans [KBD01], Kuncheva fait la différence entre la fusion et la sélection. La fusion consiste à combiner toutes les sorties, alors que la sélection consiste à choisir les "meilleurs" parmi un ensemble de classifieurs possibles pour identifier la forme inconnue.

Duin [DT00], distingue 2 types de méthodes de combinaison dans la fusion : (1) combinaison des classifieurs différents et (2) de classifieurs faibles. Cette dernière présente la même structure mais entraînée sur des données différentes.

Un autre regroupement est proposé par Xu [XKS92], qui distingue les méthodes de combinaison uniquement par le type de sortie des classifieurs (classe, mesure) présenté en entrée de la combinaison. Jain [JDM00], construit une dichotomie suivant deux critères : le type de sorties des classifieurs et leur capacité d'apprentissage. Ce dernier critère est aussi utilisé par Kuncheva [Kun03] pour séparer les méthodes de fusion. Les méthodes avec apprentissage permettent de chercher et d'adapter les paramètres à utiliser dans la combinaison suivant la base des exemples disponibles. Les méthodes sans apprentissage se contentent d'utiliser seulement et simplement les sorties des classifieurs sans intégrer d'autres informations à priori sur les performances de chacun des classifieurs.

Nous proposons une dichotomie (Fig. 3.2) qui permet de distinguer au premier niveau les méthodes de fusion. Ces dernières peuvent être séparées suivant la nature des classifieurs combinés : combinaison de classifieurs faibles ou de classifieurs différents. Si dans le premier cas, les méthodes combinent les résultats de classifieurs identiques mais entraînés sur des

données de distributions différentes, la situation est entièrement différente en ce qui concerne la deuxième méthode dans laquelle est combinée des classifieurs qui se différencient aussi bien par leur structure que par les données traitées et les caractéristiques utilisées.

Dans les méthodes de combinaison de classifieurs différents, on distingue les méthodes dites « figées » ou sans apprentissage et des méthodes avec apprentissage qui cherchent à apprendre, sur les données disponibles, les paramètres nécessaires à la combinaison. Enfin, la complexité de ces méthodes peut varier en fonction du niveau d'information associé aux réponses fournies par les classifieurs à combiner (sortie de type classe, mesure).

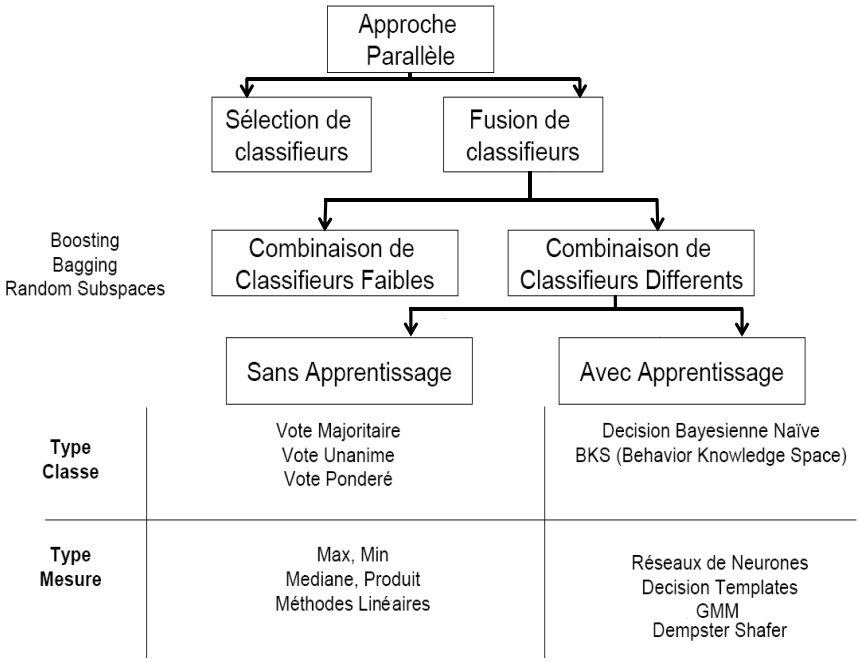


FIG. 3.2 – Dichotomie des méthodes de combinaison parallèle de classifieurs.

3.2.1.3 Combinaison de classifieurs différents

Les classifieurs peuvent être de différentes natures (e.g. combinaison entre la sortie d'un réseau de neurones et d'un SVM). Ci-dessous, nous allons décrire les méthodes avec et sans entraînement.

Méthodes sans apprentissage

Les classifieurs de type classe proposent uniquement la classe d'appartenance de l'élément à reconnaître, ainsi les seules méthodes à appliquer pour combiner ces résultats sans apprentissage sont basées sur le principe de vote. Elles sont très utilisées pour la reconnaissance de l'écriture [CTS94]. Toutes les méthodes de votes peuvent être dérivées de la règle de majorité avec seuil, exprimée par :

$$E(x) = \begin{cases} C_i \text{ si } \sum e_j \geq \alpha L \\ \text{Rejet sinon} \end{cases} \tag{3.1}$$

où L est le nombre de classifieurs à combiner.

Pour $\alpha = 1$, la classe finale est choisie si tous les classifieurs proposent cette réponse sinon il y a rejet de la réponse. Cette méthode restrictive qui n'accepte pas de risque possible est appelée *majorité unanime*. Pour $\alpha = 0.5$, cela signifie que la classe finale est décidée si plus de la moitié des classifieurs l'ont proposé, on est dans un vote à *majorité absolue*. Pour $\alpha = 0$, il s'agit de la *majorité simple* où la décision finale est la classe la plus proposée parmi les L classifieurs. Dans la *majorité pondérée*, la réponse de chaque classifieur est pondérée par un coefficient indiquant son importance dans la combinaison [AB96]. Cette dernière sera utilisée ultérieurement comme proposition pour améliorer l'entraînement (voir section 3.2.1.4).

Les classifieurs de type mesure combinent des valeurs qui reflètent le degré de confiance sur l'appartenance de l'élément à reconnaître en chacune des classes. Dans ce cas, la règle de décision est donnée par les méthodes linéaires qui consistent tout simplement à appliquer aux sorties des classifieurs une combinaison linéaire [Ho92] :

$$E(x) = \sum_{k=1}^{L} \beta_k m_i^k \tag{3.2}$$

β_k est le coefficient qui détermine l'importance attribuée au $k^{\text{ième}}$ classifieur dans la combinaison et m_i^k est la réponse pour la classe i.

Méthodes avec apprentissage

Contrairement aux techniques à base de votes, plusieurs méthodes utilisent l'étape d'apprentissage pour combiner les sorties de classifieurs, considérant la combinaison comme une étape de classification, en particulier en utilisant les méthodes probabilistes qui estiment une probabilité représentant un degré d'appartenance à une classe. Quelques méthodes ont été abordées dans la section 2.2. On citera : le réseau de neurones, le mélange de gaussienne, le classifieur K-plus proche voisins, théorie bayesienne, etc. Dans cette partie, nous allons décrire la combinaison bayesienne, suivie de trois méthodes efficaces de combinaison.

Notations :
- Soient $e = \{e_1, ..., e_L\}$ le vecteur de L sorties de classifieurs (i.e. un classifieur SVM par descripteur) ;
- $c = \{c_1, c_2, ..., c_M\}$ ensemble de M classes.

1. **Naïve bayesienne (BN)**
   Rappel :
   - La formule de Bayes : $p(A/B) = \frac{p(B/A)p(A)}{p(B)}$
   - La règle de classification de Bayes recommande de classer le vecteur e dans la classe c_k pour laquelle $P(c_k/e)$ est maximale. Ce qui revient à maximiser $\frac{p(e/c_k)p(c_k)}{p(e)}$. Soit encore $p(e/c_k)p(c_k)$, du moment ou $p(e)$ ne dépend pas de c_k.

La méthode naïve bayesienne repose sur l'hypothèse de l'indépendance entre les entrées des sources d'informations. Pour pouvoir appliquer la règle de Bayes, il faut donc pouvoir estimer $p(e/c_k) = \prod_{i=1}^{L} p(e_i/c_k)$ et $p(c_k) = \frac{n_k}{n}$. Ce dernier est calculé par l'ensemble d'entraînement (i.e. n_k nombre d'éléments d'un total de n qui appartiennent à la classe c_k).

Grâce à cette hypothèse, la règle de classification de Bayes devient alors : *classer le vecteur e dans la classe c_k qui maximise :* $p(c_k) \prod_{i=1}^{L} p(e_i/c_k)$.

Lors de la description des données, nous n'avons pas pu étudier la validité de cette hypothèse. Cette méthode de fusion ne peut se justifier.

2. **Décision Template (DT)** a été proposée par Kuncheva [KBD01, Kun03]. Comme son nom l'indique, cette technique repose sur l'utilisation de "patrons de décision". Elle utilise une approche reposant sur le calcul d'une distance à des prototypes qui nécessite un apprentissage pour la détermination du patron pour chacune des classes. Ce dernier correspond à la moyenne des sorties numériques des différentes sources pour tous les individus d'une même classe.

   Formellement, soit T_k, l'ensemble d'individus e de l'ensemble d'apprentissage appartenant à la classe c_k. Le patron de décision DT^k associé à la classe k est alors déterminé par la formule suivante :

$$DT^k = \frac{\sum_{e(i,j) \in T_k} P_j^i}{Card(T_k)} \tag{3.3}$$

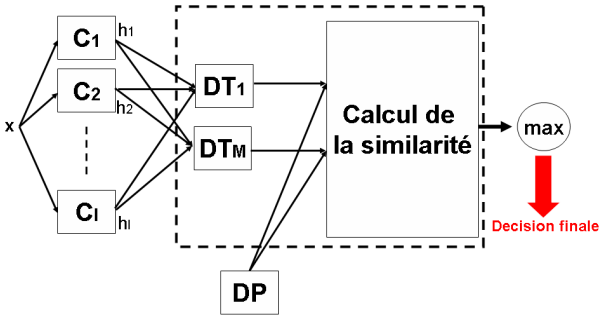


FIG. 3.3 – Schéma descriptif de la méthode Décision Template.

Un patron de décision est donc une matrice de taille $[L, M]$ avec L classifieurs et M classes. Pour effectuer la fusion des informations en disposant de M profils de décision, il reste donc à déterminer quel patron de décision est le plus similaire au profil de l'individu à classer. Le profil d'un individu étant constitué des sorties de toutes les sources, c'est à dire, dans notre cas, des probabilités à posteriori représentées par la matrice DP (*Decision Profil*).

Plusieurs mesures de similarité peuvent être envisagées. Par exemple, il est possible d'utiliser la mesure de similarité suivante :

$$Sim(DP(e_t), DT^k) = \frac{\sum_{i,j=1}^{L,M} \min(DP(e_t)_{i,j}, DT_{i,j}^k)}{\sum_{i,j=1}^{L,M} \max(DP(e_t)_{i,j}, DT_{i,j}^k)} \tag{3.4}$$

Où à partir de la distance Euclidienne.

$$Sim(DP(e_t), DT^k) = 1 - \frac{1}{LM}\sum_{i=1}^{L}\sum_{j=1}^{M}(DP(e_t)_{i,j} - DT^k_{i,j}) \tag{3.5}$$

Finalement, la décision est prise grâce au maximum des différentes similarités. Nous avons utilisé la distance Euclidienne pour tester cette méthode. D'autres normes ont été évaluées, on citera Mahalanobis et Manhattan. Les résultats de la comparaison de ces normes dans [BH06] montrent que la mesure Euclidienne est celle qui a donnée le plus de satisfaction.

3. **Les Algorithmes Génétiques (AG) :** Plusieurs opérations simples comme le minimum, le maximum et la somme peuvent être appliqué dans le domaine de la fusion, mais le choix reste arbitraire. Souvannavong et al. [Sou05] proposent d'utiliser l'arbre binaire basé sur l'algorithme génétique pour sélectionner la meilleure formule menant à la fusion. La partie suivante décrit la représentation des combinaisons possibles pour effectuer la fusion. Ensuite, nous verrons comment les algorithmes génétiques génèrent et sélectionnent les meilleures combinaisons.

Les arbres binaires pour la fusion :

Les quatre opérations retenues en premier lieu sont la somme, le produit, le maximum et le minimum. Elles ont un sens particulier dans un cadre probabiliste qui convient parfaitement au problème de la fusion.

- **La somme** permet de mettre l'accent sur la présence d'un concept dans une des modalités au moins ;
- **Le produit** valorise les concepts présents dans toutes les modalités ;
- **Le maximum** et le **minimum** permettent d'effectuer une sélection par vote.

L'objectif est de modéliser l'ensemble des combinaisons possibles en utilisant les opérations citées. Les arbres binaires sont alors une bonne solution pour représenter l'ensemble des fonctions possibles. Les feuilles correspondent aux descripteurs tandis que les noeuds identifient les opérations à mener. La structure hiérarchique implique un ordre dans le déroulement des opérations, ce qui permet d'effectuer certaines opérations en priorité (une somme avant un produit par exemple). La Fig 3.4 illustre la représentation d'une fonction par un arbre binaire.

Le nombre de configurations possibles d'un arbre binaire complet à n noeuds internes est égale au n-ème nombre de Catalan [1] C_n [Sou05]. Sachant que pour chaque noeud, quatre opérations sont proposées, le nombre de formules possibles est alors de $4^n C_n$, avec :

$$C_n = \frac{(2n)!}{n!(n+1)!} \tag{3.6}$$

[1] Nombre de Catalan est une suite de nombres que l'on rencontre souvent pour compter des objets (e.g. combien de graphes différents peut-être formés ?).

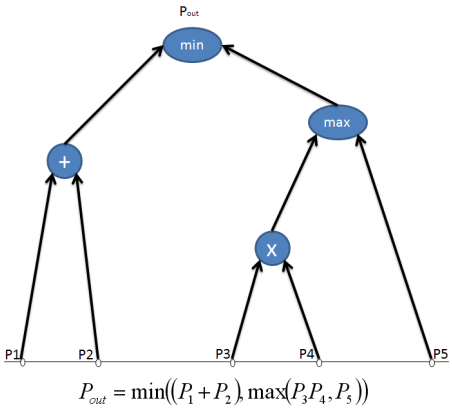


FIG. 3.4 – Représentation d'une fonction par un arbre binaire.

Etant donné le grand nombre de combinaisons possibles, il n'est pas concevable d'effectuer une recherche exhaustive de la solution. L'algorithme génétique est une méthode adaptée pour trouver efficacement la meilleure combinaison rapidement. Pour un problème d'optimisation donné, un individu représente un point de l'espace des états. On lui associe la valeur du critère à optimiser. L'algorithme génère ensuite de façon itérative des populations d'individus sur lesquelles ils appliquent des processus de *sélection, de croisement et de mutation*. La sélection a pour but de favoriser les meilleurs éléments de la population, tandis que le croisement et la mutation assurent une exploration efficace de l'espace des états.

Le mécanisme consiste à faire évoluer, à partir d'un tirage initial, un ensemble de points de l'espace vers le ou les optima d'un problème d'optimisation. L'ensemble du processus s'effectue à taille de population constante, notée n_P. Afin de faire évoluer ces populations de la génération k à la génération $k+1$, trois opérations sont effectuées pour tous les individus de la génération k :

(a) **Une sélection** d'individus de la génération k est effectuée de façon à privilégier la reproduction des bons éléments au détriment des mauvais.

(b) L'opérateur de **croisement** est appliqué avec une probabilité P_c à deux éléments de la génération k (parents) qui sont alors transformés en deux nouveaux éléments (les enfants) destinés à les remplacer dans la génération $k+1$;

(c) Certaines composantes (les gènes) de ces individus peuvent ensuite être modifiées avec une probabilité P_m par l'opérateur de **mutation**. Cette procédure vise à introduire de la nouveauté dans la population.

L'évolution de la population permet de trouver des solutions qui se raffinent avec le temps. Par nature, les algorithmes génétiques peuvent évoluer indéfiniment et mettre au jour de nouvelles solutions. Toutefois, il est préférable de définir des critères d'arrêts. Deux critères sont souvent utilisés :

– le premier détecte la stabilité du critère à optimiser au cours du temps ;
– le second impose un nombre d'itérations limité.

Au final, la théorie des algorithmes génétiques est particulièrement simple et sa mise en oeuvre est facile. Seules les fonctions de mutation et de fusion sont délicates à définir et spécifiques à chaque situation. La procédure complète est décrite dans la thèse de F. Souvannavong [Sou05].

4. **Behavior Knowledge Space (BKS) :** Contrairement aux méthodes de fusion populaire qui se basent sur l'hypothèse d'indépendance entre classifieurs, BKS s'inscrit dans la non nécessité de cette hypothèse. En effet, elle permet de prendre en compte les dépendances et les intègrent directement dans la règle de décision. Le BKS est un espace à L-dimensions ou chaque dimension correspond à la décision d'un classifieur individuel [RR03]. Elle utilise un espace de connaissance qui contient toutes les décisions des classifieurs ce qui permet de connaître leurs comportements. Le BKS est basé sur deux étapes : "l'entraînement" et la "décision". Dans la première étape, la table BKS est construite en utilisant les L classifieurs et les trois informations suivantes : le nombre total des échantillons d'entrée T_i, la classe la plus représentée R_c et le nombre d'échantillons par classe N_i^c. Ensuite, la décision est prise comme suit :

$$F(P) = \begin{cases} R_c & , \; si \; T_i > 0 \; et \; \frac{N_i^{P_c}}{T_i} \geq \lambda; \\ Rejet & , sinon. \end{cases}$$

où $\lambda \in [0, 1]$ est un seuil qui contrôle la fiabilité de la décision finale, obtenu par l'apprentissage.

Cependant, le BKS présente quelques inconvénients :
- exige des ensembles d'entraînements riches et représentatifs ;
- présente un risque de sur-apprentissage ;
- dans le cas d'un petit ensemble, beaucoup de cellules de la matrice BKS seront vides réduisant le nombre de cas d'estimation.

3.2.1.4 Combinaison de classifieurs faibles

Pour améliorer le résultat des classifieurs faibles (i.e. n'importe quel algorithme légèrement meilleur que le hasard), il est souvent utilisé une des deux premières méthodes suivantes :

1. **AdaBoost** est une méthode d'agrégation de classifieurs, non seulement la plus efficace en pratique, mais également celle qui repose sur des propriétés théoriques les plus solides. C'est une technique générale qui permet de transformer des règles de décisions grossières en une règle de décision très précise. La mise à jour adaptative de la distribution des exemples, visant à augmenter le poids de ceux mal appris par le classifieur précédent, force l'apprenant à se concentrer sur les exemples difficiles. Une relance du classifieur sur cet nouvel ensemble pondéré, permet d'améliorer les performances de n'importe quel algorithme d'apprentissage. Les hypothèses $h_i(x)$ et leurs poids α_i sont construites incrémentalement par le même classifieur de manière séquentielle. Enfin, la décision se fait en utilisant la règle :

$$H(x) = signe\left(\sum_{i=1}^{T} \alpha_i h_i(x)\right) \tag{3.7}$$

$\mathcal{S} = \{(\boldsymbol{x}_1,\boldsymbol{u}_1),\ldots,(\boldsymbol{x}_m,\boldsymbol{u}_m)\}$, avec $\boldsymbol{u}_i \in \{+1,-1\}, i = 1,m$
pour tout i=1,m **faire**
  $p_0(\boldsymbol{x}_i) \leftarrow 1/m$
fin pour
$t \leftarrow 0$
tant que $t \leq T$ **faire**
  Tirer $\mathcal{S}_t$ dans $\mathcal{S}$ selon les probabilités p_t
  Apprendre une règle de classification h_t sur $\mathcal{S}_t$ par l'algorithme $\mathcal{A}$
  Soit ε_t l'erreur apparente de h_t sur $\mathcal{S}$. Calculer $\alpha_t \leftarrow \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$
  pour $i = 1,m$ **faire**
    $p_{t+1}(\boldsymbol{x}_i) \leftarrow \frac{p_t(\boldsymbol{x}_i)}{Z_t} e^{-\alpha_t}$ si $h_t(\boldsymbol{x}_i) = u_i$ (bien classé par h_t)
    $p_{t+1}(\boldsymbol{x}_i) \leftarrow \frac{p_t(\boldsymbol{x}_i)}{Z_t} e^{+\alpha_t}$ si $h_t(\boldsymbol{x}_i) \neq u_i$ (mal classé par h_t).
      (Z_t est une valeur de normalisation telle que $\sum_{i=1}^{m} p_t(\boldsymbol{x}_i) = 1$)
  fin pour
  $t \leftarrow t + 1$
fin tant que
Fournir en sortie l'hypothèse finale : $H(\boldsymbol{x}) = \text{signe}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)$

FIG. 3.5 – Algorithme de l'Adaboost.

Néanmoins, les capacités de l'Adaboost à être immunisé contre le sur-apprentissage ont été remises en cause dès qu'il s'agit de l'appliquer à des données fortement bruitées. Cette situation est fréquente avec les bases modernes, issues des nouvelles technologies d'acquisition de données, comme le Web. La vitesse de convergence du boosting se trouve également pénalisée sur ce type de bases [FS96]. Notons aussi qu'il faut régler le paramètre T (le nombre d'itérations).

2. **Bagging** est basée sur le concept du **B**oostrapping et d'**ag**gre**ging**, conçu pour générer au hasard et avec remise K copies indépendantes de N objets $X^b = (X_1^b, ..., X_n^b)$ appelées *boostrap* à partir de l'ensemble initial des échantillons d'apprentissage. Ceci est dans le but de construire un classifieur avec chacune de ces copies. L'agrégation consiste à combiner ces classifieurs en utilisant le vote majoritaire comme règle de combinaison. Skurichina [SD98] a montré que généralement le Bagging améliore la performance des classifieurs instables [2], et détériore celle des classifieurs stables. Contrairement au Boosting, les ensembles d'apprentissage et les classifieurs sont construits de manière indépendante.

[2] Une méthode de discrimination est dite instable si un changement mineur dans les données provoque un changement assez important du modèle.

3. **Random Subspace (RS)** consiste à modifier la base d'apprentissage comme dans les deux autres techniques (Bagging et Boosting). Cependant, cette modification est réalisée sur l'espace des attributs. Dans [SD98], la méthode *RS* permet de maintenir une erreur d'apprentissage faible et de diminuer l'erreur de généralisation pour les classifieurs linéaires. On constate aussi qu'elle peut être meilleure que le bagging et le boosting si le nombre de caractéristiques est élevé.

4. **Weighted BagFolding (WBF) :** connaissant la limitation des bases de données (e.g. TRECVid) due au faible nombre d'exemples positifs, l'instabilité du Bagging [SD98] (i.e. un petit changement au niveau des données peut changer complètement le comportement du classifieur), le risque du sur-apprentissage (i.e. quand on a un nombre d'itération élevé [FS96]) et la forte sensibilité de l'Adaboost au bruit (i.e. l'une des conséquences possibles est que les exemples bruités, sur lesquels finit par se concentrer l'algorithme, perturbent l'apprentissage de l'Adaboost) ainsi que l'hypothèse qui devient trop complexe (sur-apprentissage). Nous proposons une méthode d'entraînement basée sur la technique *N-Folding* [3] et qui s'inspire du Bagging, que nous appelons *Weighted BagFolding*.

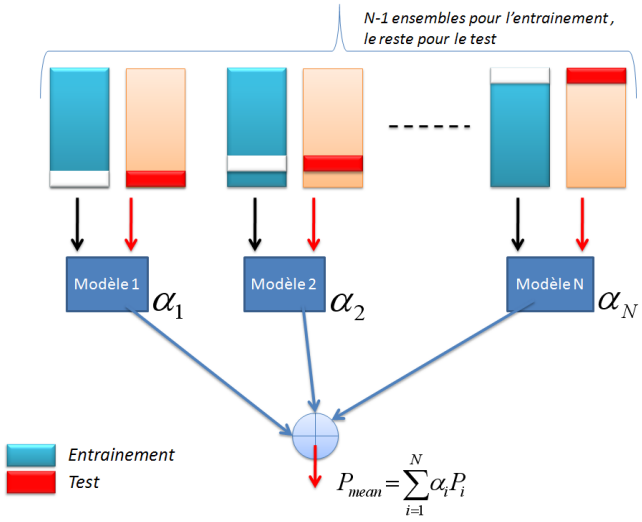


FIG. 3.6 – La combinaison par WBF.

Au lieu d'entraîner le modèle sur un échantillon bootstrap (i.e. sélection au hasard avec remise) comme le Bagging ou certaines observations sont dupliquées tandis que d'autres sont absentes, ce qui introduit une part d'aléatoire. Nous allons utiliser le principe du *N-Folding*, plus particulièrement le *Ten-Folding*. Cette dernière est utile pour évaluer la pertinence d'un type de classificateur dans le cas ou le nombre de données ne permet pas le partitionnement. Ainsi, $N = 10$ modèles pondérés par un

[3] Le principe du *N-Folding* est de diviser la base de données en N ensembles : $(N-1)$ ensembles pour l'entraînement et le reste pour le test. Ce processus va être répéter pour toutes les combinaisons possibles comme le montre la Fig. 3.6. La décision finale est donnée par la moyenne des sorties des modèles.

coefficient α_i (indique l'importance dans la combinaison) sont calculés. Ce poids est obtenu en utilisant l'erreur de validation $\epsilon_i \leq \frac{1}{2}$ (Eq. 3.8). Ce choix n'est pas arbitraire, il est obtenu par le même procédé de calcul du poids par l'Ababoost pour en pouvoir se rapprocher et nous comparer [4]. On aura alors un modèle avec un faible poids si ϵ_i est élevé et vice versa. L'intérêt de telle méthode est de répéter la procédure et d'utiliser chaque ensemble de données pour construire un modèle et le valider, tout en étant sure que toutes les données ont été utilisées. Nous disposons alors de différentes réalisations de la statistique estimée (i.e. modèle).

$$\epsilon_i = \sum_{j=1}^{N} (y(x_j) - f(x_j))^2 \tag{3.8}$$

$$\alpha_i = \frac{1}{2} \log(\frac{1 - \epsilon_i}{\epsilon_i}) \tag{3.9}$$

Enfin, la décision finale est la combinaison de toutes les mesures et de leurs poids. Ce qui rend le modèle WBF plus précis qu'un simple entraînement avec une sommation directe des résultats.

$$H(x) = \sum_{i=1}^{N} \alpha_i P_i(x) \tag{3.10}$$

3.2.2 Réseau de neurones basé sur la théorie des évidences (NNET)

Dans cette partie, nous allons voir comment la théorie des évidences peut être appliquée à la fusion de classifieurs. Un modèle basé sur la propagation de l'information par le neurone combiné avec l'idée du degré de confidence [ZD95] sera étudié, appliqué, puis une évolution de ce modèle sera réalisée.

3.2.2.1 Présentation de la théorie des évidences

Cette théorie est un formalisme très général permettant d'englober à la fois la théorie des probabilités et la théorie des possibilités. Elle repose sur la répartition d'une fonction de masse m (Basic Belief Assignment) aux différents éléments (ensembles focaux) du cadre de discernement $\Omega = \{w_1, w_2, ..., w_M\}$. La fonction de masse est une fonction de $2^\Omega = \{\emptyset, w_1, w_2, \{w_1, w_2\}, w_3, \{w_1, w_3\}, ..., \Omega\} \rightarrow [0, 1]$ qui vérifie [Sha76] :

$$\begin{cases} m(\emptyset) = 0 \\ \sum_{A \subseteq \Omega} m(A) = 1 \end{cases} \tag{3.11}$$

La fonction de masse est dite *normale* si $m(\emptyset) = 0$, est souvent interprété comme le degré de conflit entre les connaissances quantifiées par m. Dans ce cas, on dit qu'on travail dans un monde *clos*, ce qui suppose que le cadre Ω est exhaustif (i.e. la valeur réelle de y est nécessairement dans Ω). Dans le cas contraire, le monde *ouvert* considère que la valeur

[4] D'autres modèles de pondération peuvent être utilisés.

réelle de y peut ne pas être dans Ω (non-exhaustive), ainsi, la fonction de masse est dite *sous-normale*.

La masse $m(A)$ représente la partie du degré de croyance placée exactement sur la proposition A. Quand la source est complètement incertaine, alors il est impossible de différencier les hypothèses et :

$$m(\Omega) = 1 \tag{3.12}$$

Si, par contre, la source est parfaite (i.e. précise et sûre), alors il existe A_i unique $\in 2^{\Omega}$ tel que :

$$m(A_i) = 1 \tag{3.13}$$

Plusieurs fonctions sont définies à partir de la fonction de masse. Ici, nous en présenterons deux, celles que nous avons utilisé. Il s'agit de la fonction de **crédibilité** (Equ 3.14) *bel*, qui représente le degré total de croyance spécifique et justifiée en A [Sme94] (i.e. *justifiée, car ne sont pris en compte que les masses de croyance allouées à des sous-ensembles $B \subseteq A$; Spécifique, car l'élément $\emptyset$ n'est pas considéré étant un sous-ensemble de A et de $\bar{A}$*), et de la fonction de **plausibilité** (Equ 3.15) *pl*, qui constitue la borne supérieure sur le degré de croyance qui pourrait être alloué à l'hypothèse A.

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \tag{3.14}$$

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B) \tag{3.15}$$

Les deux fonctions "*pl*,*bel*" permettent la création d'un intervalle $[bel, pl]$, qui représente le domaine de variation de la probabilité associée à une hypothèse. Choisir une hypothèse sur la base de ce critère revient donc à retenir la plus certaine. Plus cet intervalle est étroit, plus l'incertitude associée à l'hypothèse est faible. Plus cet intervalle tend vers le singleton $\{1\}$ et plus l'hypothèse est probable.

La théorie des évidences permet de combiner des fonctions de masses issues de différentes sources d'informations. Les modes de combinaisons seront plus particulièrement présentés dans la section 3.4.4.

3.2.2.2 Modélisation de la fonction de masse

Maintenant, il est nécessaire d'estimer la fonction de masse. Plusieurs modèles existent [Blo03]. Le plus simple est calculé sur les singletons dans une source, souvent estimé comme une probabilité (Equ. 3.16). Les masses sur tout les autres sous-ensembles sont nulles. Il est clair que cette méthode est réductrice et n'exploite pas vraiment les avantages de la théorie des évidences. Dans ce sens, nous allons exposer deux méthodes d'estimation de la masse proposée dans la littérature [5] qui traitent de ce problème [Blo03] :

[5] Une description globale de plusieurs méthodes pour l'estimation de la fonction de masse est présentée dans [Blo03]. Nous nous sommes particulièrement intéressés à deux modèles : (1) basé sur les probabilités et (2) sur le calcul de distances.

$$\begin{cases} m(\{w_i\}) = P_i \\ m(\{A\}) = 0 \quad \forall A \in 2^{\Omega} - \{w_i\} \end{cases} \tag{3.16}$$

1. **Modèle probabiliste :** ce modèle s'appui sur la notion d'affaiblissement d'une source en fonction de sa fiabilité, en utilisant des méthodes d'apprentissage classiques. Les nouvelles masses $m^{'}$ sont calculées à partir des masses initiales m comme le montre l'équation (Equ. 3.17).

$$\begin{cases} m^{'}(\{w_i\}) = \alpha m(\{w_i\}) \\ m^{'}(\Omega) = 1 - \alpha + \alpha m(\{\Omega\}) \end{cases} \tag{3.17}$$

où $\alpha \in [0, 1]$ est un coefficient d'affaiblissement. $\alpha = 0$ si la source n'est pas fiable.

Dans le cas ou les masses sont apprises sur les singletons seulement, par exemple à partir des probabilités (i.e. $m(\Omega) = 0$), cette technique parvient à affecter une faible masse à Ω. On obtient :

$$\begin{cases} m^{'}(\{w_i\}) = \alpha m(\{w_i\}) \\ m^{'}(\Omega) = 1 - \alpha \end{cases} \tag{3.18}$$

Cependant, les disjonctions d'hypothèses ne sont pas modélisées, ce qui limite l'efficacité de ce modèle. Appriou et al [App93] proposent d'introduire une estimation initiale de probabilités conditionnelles $P(f(x)/w_i)$ (où $f(x)$ désigne les caractéristiques de l'élément x extraites de la source). On notera pour simplifier l'écriture $P(x/w_i)$ à travers deux types de modèles probabilistes. Le premier modèle est donné par la fonction de masse associée à une source via la combinaison de fonctions de masse associées à chaque singleton (Eq. 3.19) :

$$\begin{cases} m^i(\{w_i\}) = \frac{\alpha_i \beta p(x/w_i)}{1+\beta p(x/w_i)} \\ m^i(\Omega/\{w_i\}) = \frac{\alpha_i}{1+\beta p(x/w_i)} \\ m^i(\Omega) = 1 - \alpha_i \end{cases} \tag{3.19}$$

où α_i est un coefficient d'affaiblissement lié à la classe w_i. β est un coefficient de pondération des probabilités (i.e. si $\beta = 0$: le calcul de la masse prend en compte que la fiabilité de la source, $\beta = 1$: incorpore aussi les données).

Le deuxième modèle probabiliste proposé s'inscrit dans le cas où l'information est essentiellement donnée sur ce qui n'est pas w_i. Alors on obtient les formules suivantes :

$$\begin{cases} m^i(\{w_i\}) = 0 \\ m^i(\Omega/\{w_i\}) = \alpha_i(1 - \beta p(x/w_i)) \\ m^i(\Omega) = 1 - \alpha_i + \alpha_i \beta p(x/w_i)) \end{cases} \tag{3.20}$$

Enfin, la masse associée à la source est calculée par la règle de Dempster-Shafer (Equ. 3.83). Ce modèle est adapté lorsqu'on apprend en utilisant le principe de "un-contre-tous", ce qui est notre cas.

2. **Modèle à base de distance :** L'idée de départ de ce modèle est la suivante : Si nous considérons l'objet x_i de classe w_q proche de l'objet x, alors une partie de la croyance sera affectée à w_q et le reste à l'ensemble des hypothèses du cadre de discernement. Ainsi, une fonction de masse associée à chaque classe peut être définie dans deux éléments focaux seulement : $\{w_q\}$ et Ω comme le montre l'équation suivante :

$$\begin{cases} m(\{w_q\}) = \alpha\phi_q(d) \\ m(\Omega) = 1 - \alpha\phi_q(d) \\ m(A) = 0 \quad \forall A \in 2^\Omega - \{\{w_q\}, \Omega\} \end{cases} \tag{3.21}$$

où $\phi(.)$ est une fonction monotone décroissante (e.g. exponentielle, etc) vérifiant l'équation (Eq. 3.22). $d(x, x_i)$ est une distance Euclidienne entre le vecteur x et le $i^{\text{ème}}$ vecteur d'entraînement. Elle permet d'affecter une masse plus importante, lorsque x tend à ressembler à w_q. $\alpha \in [0, 1]$ est une constante, elle prévient une affectation totale de la masse à la classe w_q quand x et le $i^{\text{ème}}$ échantillon sont égaux. γ_q est un paramètre positif qui définit la vitesse de décroissance de la fonction de masse. Enfin, la masse associée à la source est calculée par la règle de Dempster-Shafer (Equ. 3.83).

$$\begin{cases} \phi_q(d) = \exp\left(-\gamma_q d^2\right) \\ \text{avec } \phi_q(0) = 1 \\ \text{et } \lim_{d\to\infty} \phi_q(d) = 1 \end{cases} \tag{3.22}$$

Dans [ZD95], ce modèle a été appliqué à l'algorithme K plus proches voisins. La distance a été calculée entre x et l'un de ses voisins.

3.2.2.3 Réseau de neurone basé sur la théorie des évidences

Après avoir exposé deux modèles d'estimation de la fonction de masse, un petit rappel de ce que nous avons déjà effectué jusqu'ici dans la conception de notre système automatique d'indexation nous semble nécessaire. A ce stade, nous avons transformé notre problème multi-classes en plusieurs problèmes bi-classes (binaire) en utilisant la technique d'apprentissage "un-contre-tous". Cela revient à classer par exemple le concept Sky contre Non-Sky (i.e. contre les 35 concepts restants). En appliquant cette technique dans la fusion, nous allons obtenir la fonction de masse suivante pour le concept Sky :

$$\mathbf{m} = [m(\{Sky\}), m(\{Non\text{-}Sky\}), m(\Omega)] \tag{3.23}$$

Dans l'évaluation des méthodes de classification dans la section 3.3, nous allons voir que l'utilisation des réseaux de neurones pour la combinaison des sorties des classifieurs donnent le meilleur taux de bonne détection comparant à l'état de l'art présenté dans ce mémoire. Connaissant les limites des méthodes probabilistes (voir la Table 3.1 comparative des méthodes), nous avons choisi d'étudier la comportement de ce genre de méthodes en utilisant la théorie des évidences. En utilisant le modèle de distance [Den00, LMJ04], où un réseau de neurones à base de la théorie des évidences a été proposé, avec une couche d'entrée L_{in}, une couche cachée L_2 et une couche de sortie L_3. Ici, nous étendons cette méthode à

notre application en introduisant une nouvelle couche comme le montre la Fig. 3.7, pour obtenir une couche d'entrée L_{in}, deux couches cachées L_2 et L_3 et une couche de sortie L_{out} que nous appelons par la sortie normalisée "Normalized output" avant la prise de décision finale. Chaque couche corresponds à une étape de la procédure suivante :

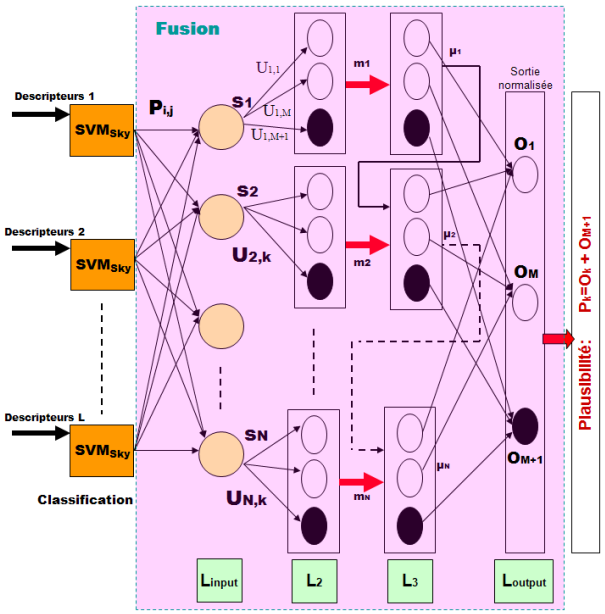


FIG. 3.7 – Représentation générale du NNET.

1. **La couche L_1 :** contient N unités (prototypes). Elle est identique à la couche d'entrée du réseau RBF avec une fonction d'activation exponentielle ϕ, et d : distance calculée par les données d'entraînement.

$$\begin{cases} s^i = \alpha^i \phi(d^i) \\ \phi(d^i) = \exp\left(-\gamma^i (d^i)^2\right) \\ \alpha^i = \frac{1}{1+\exp\left(-\epsilon^i\right)} \end{cases} \tag{3.24}$$

avec $\alpha \in [0,1]$ est un paramètre d'affaiblissement associé au prototype $i = \{1, ..., N\}$ ($\epsilon = 0$ à l'initialisation, $\gamma^i = 0.1$ est la taille de réceptivité du prototype) [BH07].

Plus formellement, le calcul est effectué uniquement par classe selon les étapes ci-dessous :

– Sélection des échantillons positifs qui représentent par exemple le concept *Sky* ;
– Calcul des Nb_c centres [6] de ce groupe en utilisant par exemple l'algorithme K-means. Les centres ainsi seront les poids de la couche d'entrée ;

[6] Il n'existe pas de méthodes directes pour optimiser ce paramètre, mais il doit être soit connu ou obtenu pendant l'entraînement. On laissera à l'apprentissage cette tache délicate. Le même constat est donné pour l'obtention du nombre de prototypes N à utiliser. Dans notre cas, on remarque que ce paramètre est plus particulièrement efficace sur cette plage $\mathrm{N} = [\frac{L}{2}, 2L]$, avec L est le nombre de classifieurs.

– Calcul de la distance Euclidienne $d = \frac{1}{2}(x - \mu_i)^2$;
– Enfin, l'introduction de l'information dans la fonction ϕ pour obtenir s ;
– Reprendre le processus avec l'autre classe *Non-Sky*.

2. **La couche L_2 :** calcule la fonction de masse m^i (Equ. 3.25) associée à chaque prototype. Elle est composée de N modules de $M+1$ (M=2 : système binaire "Classe, Non-classe") unités chacune. Les unités du module i sont connectées au neurone i de la couche précédente.

$$\begin{cases} m^i(\{w_q\}) = \alpha^i u_q^i \phi(d^i) \\ m^i(\Omega) = 1 - \alpha^i \phi(d^i) \end{cases} \tag{3.25}$$

avec, u_q^i représente le degré du lien entre le prototype et chaque classe w_q (q : index des classes $q = \{1, 2\}$). Ici, l'idée est d'initialiser le vecteur u par le vecteur classes des centres obtenus. Par exemple, on aura pour le premier prototype la matrice u suivante :

$$u = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

3. **La couche L_3 :** la règle de combinaison de Dempster-Shafer combine N différentes fonctions de masses en une seule masse. Grâce aux propriétés de commutativité et d'associativité de la combinaison conjonctive, le réseau est indépendant de l'ordre des prototypes. Dans [Den00], les vecteurs d'activation du module i sont définies comme :

$$\begin{cases} \mu^i = \oplus_{k=1}^{i} m^k = \mu^{i-1} \oplus m^i \\ \mu^1 = m^1 \end{cases} \tag{3.26}$$

Le vecteur d'activation pour $i = \{2, ..., N\}$ peut être calculé récursivement par :

$$\begin{cases} \mu^1 = m^1 \\ \mu_j^i = \mu_j^{i-1} m_j^i + \mu_j^{i-1} m_{M+1}^i + \mu_{M+1}^{i-1} m_j^i \\ \mu_{M+1}^i = \mu_{M+1}^{i-1} m_{M+1}^i \end{cases} \tag{3.27}$$

Notons $M+1$ par Ω. En développant l'équation par prototype, il est clair que $\mu_j^i = \mu_j^{i-1} m_j^i + \mu_j^{i-1} m_\Omega^i + \mu_\Omega^{i-1} m_j^i$ ajoute dans chaque itération de l'ignorance (Fig. 3.8), l'ignorance du prototype précédent $i-1$ à celui de i.

4. **La couche de sortie L_{out} :** Dans [Den00], la sortie du réseau est directement obtenue par $O_j = \mu_j^N$. Or, les expérimentations montrent que si l'ordre des prototypes n'affecte pas les résultats, cette sortie directe est assez sensible au nombre de prototype choisi, ou une petite modification dans le nombre peut complètement changer le comportement du fusionneur. Afin de résoudre ce problème, et dans l'idée de baisser la quantité d'ignorance qui s'accumule donnant ainsi un système très prudent et qui

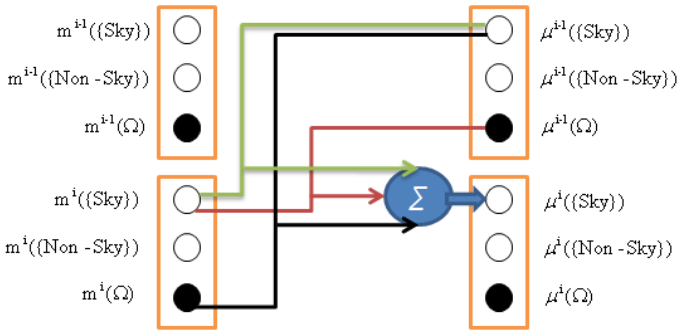


FIG. 3.8 – Représentation de la combinaison de Dempster-Shafer entre deux prototypes.

rejette la prise de décision dans nos singletons, on va ajouter une couche de normalisation et de décision (Eq. 3.28). Ici, la sortie est calculée en prenant en compte les autres vecteurs d'activations de chaque prototype afin de diminuer l'effet d'un éventuel comportement négatif du système.

$$O_j = \frac{\sum_{i=1}^{N} \mu_j^i}{\sum_{i=1}^{N} \sum_{j=1}^{M+1} \mu_j^i} \tag{3.28}$$

Les différents paramètres (Δs, Δu, $\Delta\alpha$, $\Delta\lambda$, $\Delta\beta$, $\Delta\gamma$,Δp) peuvent être déterminés par la méthode de descente de gradient de l'erreur. Enfin, la prise de décision sera obtenue par le calcul du maximum de plausibilité P_q (i.e. le degré de croyance maximum qui pourrait être alloué à l'hypothèse, d'ou la prise en compte de O_Ω [7]) de chaque classe w_q comme le montre la formule suivante :

$$P_q = \begin{cases} \text{Rejet} \quad si \quad O_\Omega \geq 0.5 \\ O_q + O_\Omega \quad \text{sinon} \end{cases} \tag{3.29}$$

Entraînement des paramètres

Les paramètres peuvent être entraînés de la même façon que le réseau RBF, par une optimisation d'un critère de performance comme la moyenne globale de l'erreur ζ produite entre le vecteur réel Y et le vecteur de décision P sur l'ensemble des l échantillons d'entraînement (Eq. 3.30). Pour notre cas, on a un système binaire $|M| = 2, M = \{1 : \text{Classe}, 0 : \text{Non Classe}\}$, il suffit de calculer cette erreur sur l'un des deux.

$$\zeta = \frac{1}{2} \sum_{s=1}^{M} (P_s - Y_s)^2 \tag{3.30}$$

[7] La prise en compte de O_Ω dans le calcul de P_q nous donne un système peu prudent si l'ignorance est élevé. Le cas contraire est observé par l'utilisation du maximum de la crédibilité $P_q = O_q$. Cette dernière n'exploite pas l'information contenue dans O_Ω, le système devient ainsi trop prudent. Le calcul du maximum de la probabilité pignistique défini par $P_q = O_q + \frac{O_\Omega}{M}$ peut être un compromis, qui consiste à partager équitablement chaque masse de croyance entre les atomes composant A. Selon le type de médias utilisés et les objectifs à atteindre, chacune des décisions pourra se justifier.

Les paramètres $s^i, u^i, \alpha^i, \lambda^i, \sigma^i, \beta^i$ peuvent être obtenus par le processus de minimisation itérative de ζ vers un minimum local de l'erreur par descente de gradient [Den00].

Ci-joint, un rappel des notations et de certaines lois de dérivation.

$$\begin{cases} d = ||x - p^i|| \\ \alpha^i = \frac{1}{1+\exp(-\xi^i)} \\ s^i = \alpha^i \exp((\sigma^i d^i)^2) \\ \mathbf{m} = m^1 \oplus_{i=2}^{N} m^i = m^1 \oplus \overline{m^i} \\ m^i_j = (m^i(\{w_1\}), m^i(\{w_2\}), m^i(\Omega)) = (m^i_1, m^i_2, m^i_\Omega) \end{cases} \tag{3.31}$$

Deux règles de dérivation seront utilisées :

$$\begin{cases} (\exp(f(x)))' = (f(x))' \exp(f(x)) \\ (\frac{f(x)}{g(x)})' = \frac{(f(x))' g(x) - (g(x))' f(x)}{(g(x))^2} \end{cases} \tag{3.32}$$

Premièrement, il est facile de remarquer que la fonction $s^i = \frac{1}{1+\exp(-\xi^i)} \exp((\sigma^i||x - p^i||)^2)$ est celle qui présente le plus de paramètres, il suffit alors de transformer la dérivation par rapport à un paramètre, vers une dérivation par rapport à s^i, pour obtenir les fonctions suivantes de mise à jour :

$$\frac{\partial \zeta(x)}{\partial \sigma^i} = \frac{\partial \zeta(x)}{\partial s^i} \frac{\partial s^i}{\partial \sigma^i} = \frac{\partial \zeta(x)}{\partial s^i} (-2\sigma^i (d^i)^2 s^i) \tag{3.33}$$

$$\frac{\partial \zeta(x)}{\partial \epsilon^i} = \frac{\partial \zeta(x)}{\partial s^i} \frac{\partial s^i}{\partial \epsilon^i} = \frac{\partial \zeta(x)}{\partial s^i} \exp(-(\sigma^i d^i)^2)(1 - \alpha^i)\alpha^i \tag{3.34}$$

$$\frac{\partial \zeta(x)}{\partial p^i_j} = \frac{\partial \zeta(x)}{\partial s^i} \frac{\partial s^i}{\partial p^j_j} = \frac{\partial \zeta(x)}{\partial s^i} (2(\sigma^i)^2 s^i (x_j - p^i_j)) \tag{3.35}$$

On a besoin de calculer $\frac{\partial \zeta(x)}{\partial s^i}$. Sachant que m est une combinaison conjonctive (Eq. 3.27), l'application au système donnera :

$$\begin{cases} m_j = m^i_j(\overline{m^i_j} + \overline{m^i_\Omega}) + m^i_\Omega \overline{m^i_j} \\ m_\Omega = m^i_\Omega \overline{m^i_\Omega} \end{cases} \tag{3.36}$$

alors,

$$\frac{\partial \zeta(x)}{\partial s^i} = \sum_{j=1}^{M} \frac{\partial \zeta(x)}{\partial P_j} \frac{\partial P_j}{\partial s^i} \tag{3.37}$$

$$= \sum_{j=1}^{M} (P_j - Y_j)(\frac{\partial m_j}{\partial s^i} + \frac{\partial m_\Omega}{\partial s^i}) \tag{3.38}$$

$$= \sum_{j=1}^{M} (P_j - Y_j)(u^i_j(\overline{m}^i_j + \overline{m}^i_\Omega) - \overline{m}^i_j - \overline{m}^i_\Omega) \tag{3.39}$$

A ce niveau, il suffit juste de remplacer cette dernière dans les équations précédentes (Eq. 3.33, 3.34, 3.35) pour obtenir les équations de mise à jour des paramètres.

Deuxièmement, la dérivée de ζ par rapport à β_j^i est donnée par :

$$\frac{\partial \zeta(x)}{\partial \beta_j^i} = \sum_{j=1}^{M} \frac{\partial \zeta(x)}{\partial u_j^i} \frac{\partial u_j^i}{\partial \beta_j^i} \tag{3.40}$$

Calculons alors $\frac{\partial \zeta(x)}{\partial u_j^i}$

$$\frac{\partial \zeta(x)}{\partial u_j^i} = \frac{\partial \zeta(x)}{\partial m_k} \frac{\partial m_k}{\partial u_j^i} = (P_j - Y_j) \frac{\partial m_k}{\partial u_j^i} \tag{3.41}$$

ainsi,

$$\frac{\partial m_j}{\partial u_j^i} = s^i(\overline{m_j^i} + \overline{m_\Omega^i}) \tag{3.42}$$

pour obtenir,

$$\frac{\partial \zeta(x)}{\partial u_j^i} = (P_j - Y_j) s^i(\overline{m_j^i} + \overline{m_\Omega^i}) \tag{3.43}$$

Cela dit, notre travail présente une normalisation de la sortie de plausibilités introduisant l'ignorance, comme le montre l'équation (Eq. 3.28), donnant $Pn_j = \frac{m_j + m_\Omega}{K}$. Avec, $K = \sum_{i=1}^{M+1} m_k$. Pour obtenir les paramètres précédents, il suffit de remplacer P_j dans l'équation (Eq. 3.30) par la valeur normalisée Pn_j, ainsi, seules les équations suivantes demanderons une dérivation différente, ce qui change celles des mises à jour, pour obtenir :

$$\frac{\partial \zeta_n(x)}{\partial u_j^i} = \frac{\partial \zeta_n(x)}{\partial m_j} \frac{\partial m_j}{\partial u_j^i(x)} \tag{3.44}$$

$$\frac{\partial \zeta_n(x)}{\partial m_j} = \frac{\partial \zeta_n(x)}{\partial Pn_j} \frac{\partial Pn_j(x)}{\partial m_j} \tag{3.45}$$

$$= \sum_{k=1}^{M} (Pn_k - Y_k)\left(\frac{\partial m_{n_k}}{\partial m_j} + \frac{\partial m_{n_\Omega}}{\partial m_j}\right) \tag{3.46}$$

$$= -\frac{1}{K^2} \sum_{k=1}^{M} (Pn_k - Y_k) P_k + \frac{1}{K}(Pn_j - Y_j) \tag{3.47}$$

$$\frac{\partial \zeta_n(x)}{\partial u_j^i} = \frac{s^i(\overline{m_j^i} + \overline{m_\Omega^i})}{K} \left((Pn_j - Y_j) - \frac{1}{K} \sum_{k=1}^{M} (Pn_k - Y_k) P_k \right) \tag{3.48}$$

Schéma	Architecture	Entraînement	Commentaires
Méthodes de votes	Parallèle	Non	Suppose l'indépendance des classifieurs.
Somme, Médiane, Moyenne	Parallèle	Non	Robuste ; Suppose l'indépendance des classifieurs.
Produit Min, Max	Parallèle	Non	Suppose l'indépendance des classifieurs.
Pondération Adaptative	Parallèle	Oui	Explore une expertise locale.
Bagging	Parallèle	Oui	Demande plusieurs classifieurs à comparer. repose sur le boostrapping.
Boosting	Parallèle Hiérarchique	Oui	Risque de sur-apprentissage ; Sensible aux données ; Demande plusieurs classifieurs à comparer.
Random Subspace	Parallèle	Oui	Demande plusieurs classifieurs à comparer (sur les attributs).
Weighted BagFolding	Parallèle	Oui	Itérative ; Processus d'entraînement long.
SVM	Parallèle	Oui	Dépend de l'échelle ; Itérative ; Entraînement long ; Non-linéaire ; Insensible au sur-apprentissage ; Bonne généralisation des performances.
Réseau de Neurones	Parallèle	Oui	Sensible à l' entraînement ; Entraînement long ; Sensible au sur-apprentissage ; Fonction d'activation non-linéaire.
GMM	Parallèle	Oui	Sensible à l'estimation des paramètres.
Décision Template	Parallèle	Oui	Sensible aux Templates et à la métrique ; Repose sur le choix de la norme.
Algorithme Génétique	Hiérarchique	Oui	Procédure itérative ; Sensible au sur-apprentissage.
Behavior Knowledge Space	Parallèle	Oui	Demande un grand ensemble d'entraînement ; Risque de sur-apprentissage ; Ne nécessite aucune dépendance entre classifieurs.
Théorie des évidences	Parallèle	Oui	Fusionne des fonctions de masses ; Plusieurs lois de combinaisons ; Introduit la notion d'ignorance.

Tab. 3.1 – Tableau comparatif des méthodes de fusion de classifieurs.

3.2.3 Perplexity-based Evidential Neural Network (PENN)

Chaque concept du LSCOM-lite (Large-Scale Concept Ontology for Multimedia) [NKFH98] est mieux représenté ou décrit par un certain ensemble de descripteurs. Intuitivement, les descripteurs de couleurs peuvent être plus performant pour certains concepts tels que Sky, Snow, Waterscape, Vegetation et moins discriminant pour Studio et Meeting.

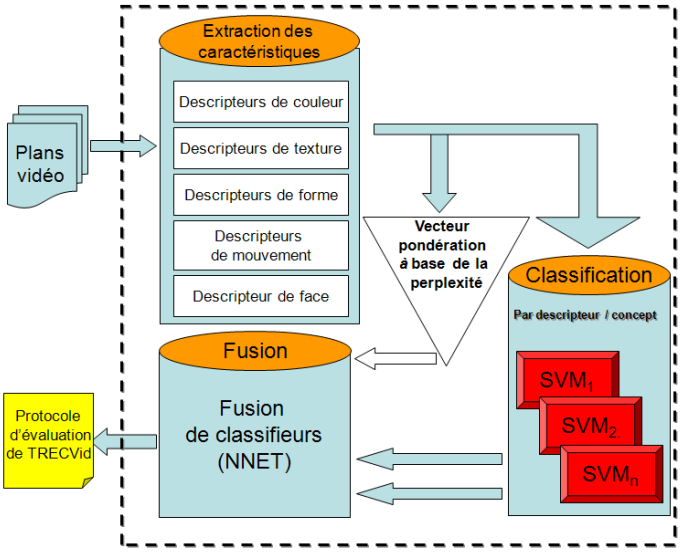


Fig. 3.9 – Schéma global du système d'indexation introduisant le PENN.

Pour cela, on propose de pondérer chaque descripteur bas-niveau selon son degré de discriminance du concept, afin d'éviter une sélection des descripteurs (Fig. 3.9). L'objectif premier de notre travail est la construction d'un système générique prêt à répondre à toutes les situations. La technique d'amélioration proposée consiste à donner plus d'importance aux caractéristiques appropriées et d'affaiblir celles qui ne le sont pas. Une solution à ce problème nous renvoie à l'entropie [Rio07]. Cette dernière mesure la quantité de l'information et de l'incertitude dans une distribution. Le principal intérêt de ce choix repose sur son pouvoir discriminant individuel des descripteurs sans faire appel à une phase d'apprentissage. Par ailleurs, nous avons adopté l'indicateur proposé par El-Yacoubi [EYGSS99] pour la reconnaissance de l'écriture : *la perplexité*. Cet indicateur est basé sur la notion d'entropie de la théorie de l'information. Puis, un vecteur de pondération est modélisé en utilisant les différentes fonctions d'évolution. Ce vecteur est ensuite combiné avec les sorties des classifieurs pour produire une nouvelle entrée qui prend en compte la relation descripteur/concept dans le NNET. La combinaison des deux processus donne ce que nous appelons "Perplexity-based Evidential Neural Network (PENN)". Comme présenté dans la Fig. 3.10, on définit maintenant les six étapes de notre méthode :

1. **K-means Clustering** calcule les k centroïdes de chaque descripteur, dans le but de créer un "dictionnaire visuel" de plans (les expérimentations montrent que $k = 2000$ présente un compromis entre l'efficacité et le temps de calcul pour le cas des

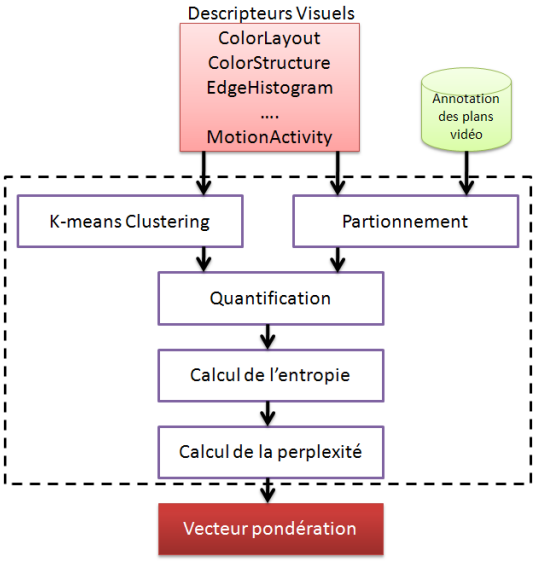


FIG. 3.10 – Etapes de calcul du vecteur de pondération (Weight) représentant la relation descripteurs/concepts.

descripteurs MPEG-7 locaux, et entre 50 à 100 pour les globaux).

2. **Partitionnement** sélectionne les échantillons positifs de chaque concept.
3. **Quantification** calcule une distance Euclidienne entre chaque partition de concept et le dictionnaire visuel. La distance minimale renvoie à incrémenter le nombre d'échantillons appartenant au centre correspondant.
4. **La mesure d'entropie :** l'entropie H (Eq. 3.49) d'une certaine distribution de descripteur $P = (P_0, P_1, ..., P_{k-1})$ donne une mesure de distribution des concepts autour des centres k [LMO04]. Dans [KS06], un bon modèle est celui où la distribution est concentrée seulement autour d'un petit nombre de centres, donnant une faible valeur d'entropie.

$$H = -\sum_{i=0}^{k-1} P_i \log(P_i) \tag{3.49}$$

où P_i est la probabilité du cluster i dans le vecteur quantifié, où la somme est calculée sur l'ensemble des mots du vocabulaire.

Dans [EYGSS99], l'auteur définit l'entropie conditionnelle des classes C considérées lors de la modélisation étant donnée un descripteur f_j par :

$$H(C/f_j) = -\sum_{i=0}^{M-1} P(c_i/f_j) \log(P(c_i/f_j)) \tag{3.50}$$

où c_i est une des classes considérées lors de la modélisation et M le nombre de ces classes. La fonction $H(C/f_j)$ permet de quantifier la capacité d'un descripteur f_j à faire la distinction entre les classes c_i. Elle atteint son maximum $log(M)$ lorsque :

$$P(c_i/f_j) = \frac{1}{M} \quad \forall i \tag{3.51}$$

Dans ce cas particulier, le descripteur f_j n'apporte aucune information pour effectuer la discrimination entre les M classes.

5. **La mesure de perplexité :** Dans [GGLL01], la perplexité PPL où la perplexité normalisée $\overline{PPL}$ (Eq. 3.52) peuvent être interprétées comme le nombre de centres dont on a besoin pour un codage optimal des données. L'avantage d'utiliser la perplexité plutôt que l'entropie est qu'elle varie entre 1 et k. Elle peut donc être directement reliée au nombre de centres mis en jeu.

$$\overline{PPL} = \frac{PPL}{PPL_{max}} = \frac{2^H}{2^{H_{max}}} \tag{3.52}$$

Si on suppose que les k centres sont distribués uniformément, on obtient alors $H(P) = \log(k)$, ainsi, $1 \leq PPL \leq k$ (ou, $\frac{1}{k} \leq \overline{PPL} \leq 1$).

6. **le vecteur pondération :** Dans les domaines de la reconnaissance de la parole, la reconnaissance de l'écriture et la correction d'orthographe [GGLL01], il est généralement conclu qu'une faible perplexité/entropie corrèle avec de meilleure performance, ou dans notre cas, une distribution fortement concentrée. Ainsi, le poids relatif du descripteur correspondant doit être augmenté. Plusieurs formules peuvent être utilisées pour représenter le poids, on citera ici la fonction Sigmoïde, Softmax, Gaussien, etc. Dans nos travaux de thèse, nous avons choisi le modèle d'évolution de Verhulst (Eq. 3.53) qui est un parent de la fonction Sigmoïde. Le modèle de Verhulst présente les avantages suivants :
   - c'est une fonction non-exponentielle,
   - avec une capacité de réception K,
   - présente un paramètre de ralentissement (*brake rate*) α_i,
   - β_i définit la rapidité de décroissance de la fonction de pondération.

$$w_i = K \frac{1}{1 + \beta_i \exp\left(-\alpha_i (1/\overline{\mathrm{PPL}_i})\right)} \tag{3.53}$$

$$\beta_i = \begin{cases} K \exp\left(-\alpha_i^2\right) & \text{si} \quad Nb_i^+ < 2 * k \\ 1 & \text{ailleurs} \end{cases} \tag{3.54}$$

β_i est introduit pour diminuer l'effet négatif produit par la limitation de notre ensemble d'entraînement, due au faible nombre d'échantillons positifs ($Nb_i^+ << k$) de certains concepts comme Weather, Desert, Mountain, etc (voir Table 1.5). On observe une faible valeur de perplexité pour ces derniers, qui ne peuvent être interprétés comme possédant une forte relation entre le descripteur et le concept. Pour éviter

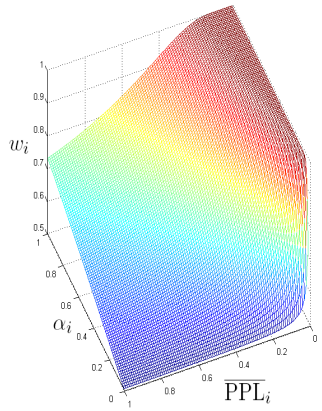


FIG. 3.11 – Modèle d'évolution de Verhulst.

cette situation, on augmente β_i (Eq. 3.54) pour obtenir une décroissance rapide du poids, pour chaque concept présentant moins de $2 * k$ échantillons positifs [8].

En résumé de cette partie, nous avons traité de la fusion de descripteurs haut-niveau. Un état de l'art a été établi. L'objectif est de mieux utiliser l'information issue des différents systèmes de classification. La théorie des évidences a été étudiée pour conclure sur une extension de la méthode initiale NNET, avec l'introduction des relations entre descripteurs et concepts pour donner le PENN. La prochaine partie évaluera les résultats des expériences conduites dans le cadre du projet NoE K-Space, afin de valider nos systèmes.

3.3 Evaluation

Dans cette étude, nous nous plaçons dans le contexte de la recherche d'information, ainsi, les mesures appropriées sont donc mises en oeuvre. Nous affranchissons les systèmes de classification d'établir une décision binaire sur l'appartenance à une classe. Il suffit alors d'ordonner les plans, de calculer les mesures de précision et de rappel. La précision sur 2 000 plans est retenue comme mesure d'évaluation afin de faciliter les comparaisons entre les différents systèmes. Toutefois, une valeur unique est parfois préférable à une courbe, en effet deux courbes peuvent être difficiles à comparer. Dans ce cas, une précision moyenne est calculée par requête, puis une moyenne de cette valeur est calculée sur plusieurs requêtes. Cette valeur unique permet une comparaison simple et rapide des performances des différents systèmes. Notons que le dernier concept qui apparaît sur l'ensemble des figures, correspond à la performance moyenne sur l'ensemble des classes (MAP : Mean Average Precision).

Un système de recherche de plans/documents peut répondre à une requête selon le tableau suivant :

[8] Le choix du seuil $2 * k$ a été obtenu par l'observation du comportement de notre système, d'autres seuils ont été testés et notre choix est celui qui donne les meilleurs résultats.

TAB. 3.2 – Tableau de contingence.

	Pertinents	Non-Pertinents	Total
Nombre de plans Retrouvés	a	b	a+b
Nombre de plans Non Retrouvés	c	d	c+d
	a+c	b+d	a+b+c+d=N

1. **la précision :** mesure la proportion de documents trouvés et corrects parmi tous les documents trouvés.
$$\text{P} = \frac{\text{Nombre de plans pertinents retrouvés}}{\text{Nombre de plans retournés}} = \frac{a}{a+b} \tag{3.55}$$
2. **le rappel :** mesure la proportion de documents trouvés et corrects parmi tous les documents indexés et corrects.
$$\text{R} = \frac{\text{Nombre de plans pertinents retrouvés}}{\text{Nombre de plans pertinents dans le corpus}} = \frac{a}{a+c} \tag{3.56}$$

Ces deux notions reflètent le point de vue de l'utilisateur : Si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des informations qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à une information qu'il souhaiterait avoir.

Par ailleurs, nous allons introduire de nouvelles mesures qui vont nous aider à une meilleure analyse et compréhension du comportement des systèmes de recherches.

3. **F-mesure/F-score :** c'est une mesure populaire qui combine la précision et le rappel, comme le montre l'équation (Eq. 3.57). Elle ne prend pas en compte la pertinence et fonctionne sur un modèle binaire : une réponse est bonne ou fausse. Ainsi, la F-mesure fournit un résultat global.
$$\text{F-mesure} = 2\frac{P.R}{P+R} \tag{3.57}$$
4. **Taux de classification des positifs (CR^+) :** mesure la proportion de documents pertinents trouvés parmi tous les documents pertinents de la base de test.
$$CR^+ = \frac{h}{g+h} \tag{3.58}$$

5. **Taux d'erreur "Balanced error rate" (BER)** est la moyenne des erreurs dans chaque classe sur un ensemble de test, dans le but de représenter correctement les gains ou les pertes du système.
$$\text{BER} = \frac{1}{2}\left(\frac{f}{e+f} + \frac{g}{g+h}\right) \tag{3.59}$$

Ces mesures nous semblent suffisantes pour l'évaluation. Cependant, d'autres mesures découlent du tableau de contingence existes qui n'ont pas été choisi dans le cadre de notre étude, comme par exemple : la pertinence = $\frac{a+d}{N}$, le taux de chute = $\frac{b}{b+d}$, le silence = $\frac{c}{a+c}$, la spécificité = $\frac{d}{b+d}$, le bruit = $\frac{b}{a+b}$, l'overlap = $\frac{a}{a+b+c}$ et la généralité = $\frac{a}{N}$.

TAB. 3.3 – Représentation des résultats par une matrice de confusion.

		Prédiction	
		Classe 0	Classe 1
Classe	Classe 0	e	f
Réelle	Classe 1	g	h

3.3.1 Evaluation de la fusion haut-niveau

Avant d'évaluer la fusion, il est intéressant de s'attarder sur le comportement de notre classification par les SVMs sur les données TRECVid'05, en utilisant indépendamment quatre descripteurs : HsvHistogram, GaborTexture, CameraMotion, EdgeHistogram. Dans la Fig. 3.12, on remarque que EdgeHistogram (EDH) est le descripteur le plus pertinent à l'ensemble des concepts, caractérisé par des contours, particulièrement pour les concepts CAR, MAPS, MOUNTAIN, SPORTS, WATERSCAPE sauf pour US FLAG où la couleur et la texture semblent être plus adaptées. Contrairement à CameraMotion qui influence peu de concepts, obtenant une performance identique au reste des descripteurs pour le concept WALKING/RUNNING.

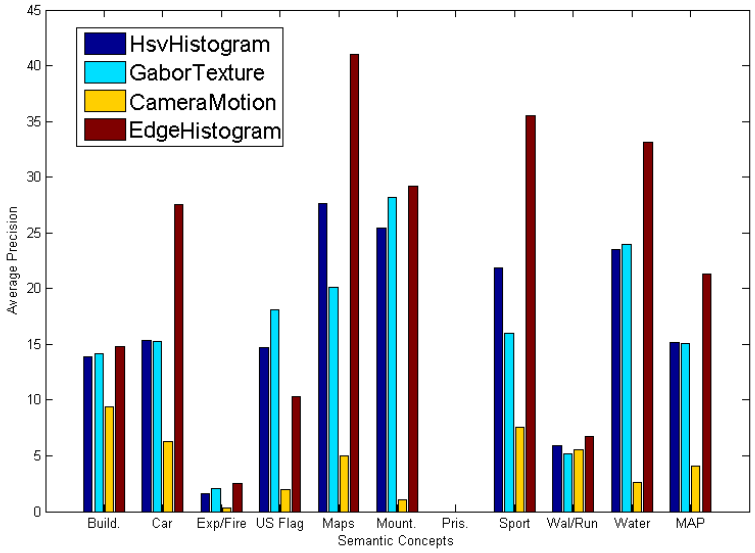


FIG. 3.12 – Performance de la classification SVM par concept, pour les quatre descripteurs.

Chaque source d'information étant en général imparfaite et insuffisante, il est important d'en combiner plusieurs afin d'avoir une meilleure connaissance. Pour étudier l'effet de la fusion sur notre système, nous allons tester dans un premier temps les méthodes décrites dans l'état de l'art dites méthodes de combinaison avec entraînement, utilisant l'algorithme

génétique (GA), mélanges de gaussiennes (GMM), Décision Templates (DT) et le réseau de neurones (NN). Les résultats sont présentés dans la Fig 3.13.

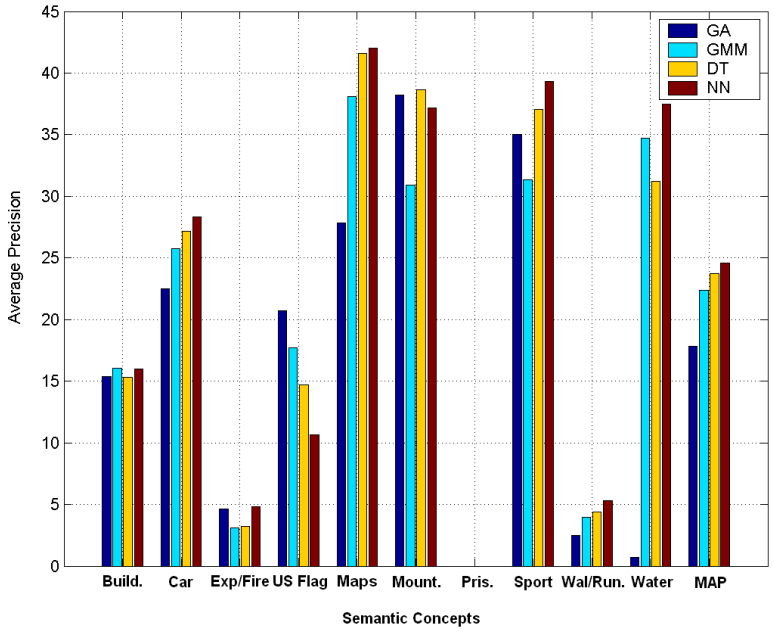


FIG. 3.13 – Comparaison de performances entre les méthodes de fusion de classifieurs.

L'algorithme GA produit des scores particulièrement faibles due à un effet de sur-apprentissage sur les concepts MAPS, WATERSCAPE. Le GA ne parvient pas à fusionner correctement les quatre précédentes sorties de classifieurs sauf pour BUILDING, EXPLOSION/FIRE, US FLAG, MOUNTAIN. Apparemment, il est plus sensible aux données bruitées comme celle de MotionCamera et s'exprime mieux lorsque EdgeHistogram n'est pas trop élevé. Dans ce cas, le résultat global du MAP_{GA} chute par rapport à celui obtenu par SVM_{EDH}. Inversement, le NN obtient des performances plus élevées sur presque tous les concepts, donnant un $\text{MAP}_{NN} = 24.85\%$. Des résultats respectables avec de faibles améliorations sont obtenus par DT et GMM mais restent inférieurs à ceux du NN. Enfin, on obtient une précision nulle pour le concept PRISONER due à la limitation de la base de donnée. Il n'y a aucun plan vidéo qui représente ce concept dans la base de test TRECVid'05 (voir Table 1.3). A ce stade, on peut déjà identifier que la fusion de classifieurs sur l'ensemble des modalités à jouer un rôle positif dans la détection de la majorité des concepts, en tirant profits de l'ensemble des informations fournies.

Concernant l'évaluation de la combinaison de classifieurs faibles, nous allons comparer maintenant les méthodes suivantes : Adaboost, Bagging, Ten-Folding (TF) avec la méthode proposée WBF. Elles sont employées pour améliorer les performances du GMM et du NN, en les considérant comme faibles. Comme le montre la Fig. 3.14, en moyenne pour tous les concepts, le WBF est celui qui présente le plus d'améliorations signifiantes sur

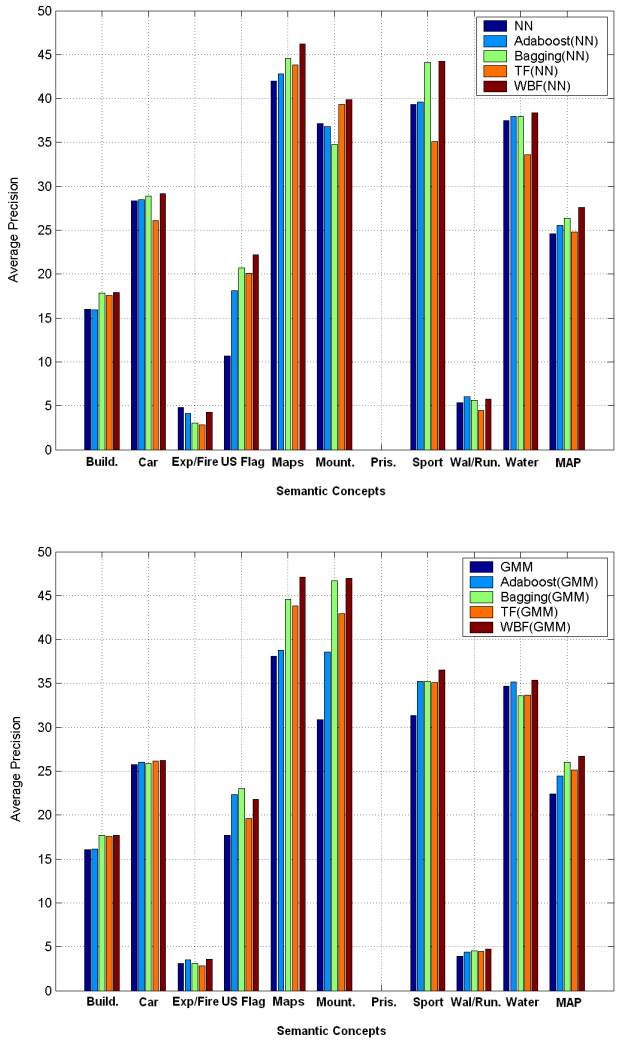


FIG. 3.14 – Performances de l'Adaboost, Bagging, Ten-Folding et le WBF pour les classifieurs faibles NN et GMM.

US FLAG, MAPS, MOUNTAIN, SPORTS et sur la précision moyenne ($MAP_{NN} = 27.63\%$, $MAP_{GMM} = 26.17\%$), malgré les limitations des données. Plus particulièrement, le WBF a réussi à doubler la précision du NN pour US FLAG et d'éliminer la perte observée dans la Fig. 3.13. Ceci est expliqué par l'apport du calcul des poids représentatifs sur le petit ensemble de validation indépendamment de l'ensemble d'entraînement, qui permet de donner plus ou moins d'importance à chaque modèle dans la combinaison, contrairement à l'Adaboost, où globalement les exemples bruités, sur lesquels finit par se concentrer le perturbent. Le Bagging, quant a lui, arrive juste derrière le WBF.

Cela dit, le WBF présente un temps de calcul plus élevé que les autres combinaisons

(i.e. chaque classifieur faible prend entre 4 à 7 minutes par concepts, ainsi 2 à 4 heures en moyenne pour les 36 concepts), mais reste une bonne méthode pour faire évoluer la précision. Cette méthode a été proposée dans ce cas particulier, mais ne sera pas utilisée dans la suite de nos travaux. Nous avons fait le choix d'opter pour un modèle plus rapide vu la grande taille des données et le nombre de concepts à traiter.

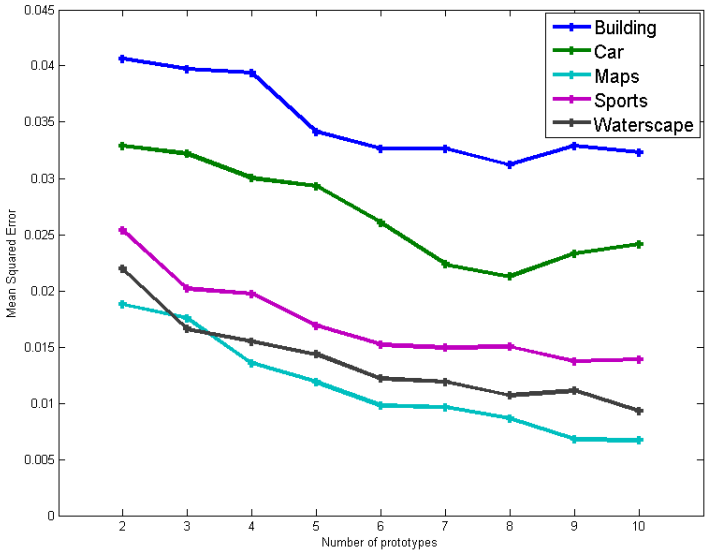


FIG. 3.15 – Variation de l'erreur d'entraînement MSE pour le NNET, en fonction du nombre de prototype utilisé, pour les concepts BUILDING, CAR, MAPS, SPORTS et WATERSCAPE.

Une deuxième série d'expériences a été conduite sur une forme particulière de réseau de neurones (réseau RBF), en comparaison avec la nouvelle méthode de fusion basée sur la théorie des évidences (NNET). Le réseau RBF et NNET ont été formé avec le même algorithme d'optimisation (la descente de gradient). Le nombre N de prototypes (neurones) a été varié entre 2 et 10, pour 4 vecteurs d'entrés. La Fig. 3.15 présente la variation de l'erreur MSE (Mean Squared Error) en fonction du nombre de prototypes choisis, pour 5 exemples de concepts "BUILDING, CAR, MAPS, SPORTS, WATERSCAPE". On observe une baisse de l'erreur avec l'augmentation du nombre de prototype, en contre partie, un accroissement du temps de calcul pour les grandes valeurs. En moyenne, l'erreur MSE d'entraînement la plus faible a été obtenue pour $N = 8$. Par ailleurs, cette expérience nous procure un premier aperçu sur la difficulté (taux d'erreur) de détection entre les 5 concepts. Les résultats de la phase d'entraînement indiquent que le concept MAPS est le plus simple à détecter, suivi de WATERSCAPE, SPORTS, CAR, puis BUILDING.

La Fig. 3.16 présente les résultats de la précision moyenne du réseau RBF et NNET. Dans cette dernière, plusieurs points peuvent être relevés : Premièrement, NNET améliore la détection de tous les concepts sémantiques, alors que le réseau RBF donne plus au moins les mêmes performances dans notre cas, qu'un simple MLP moyennant un temps d'appren-

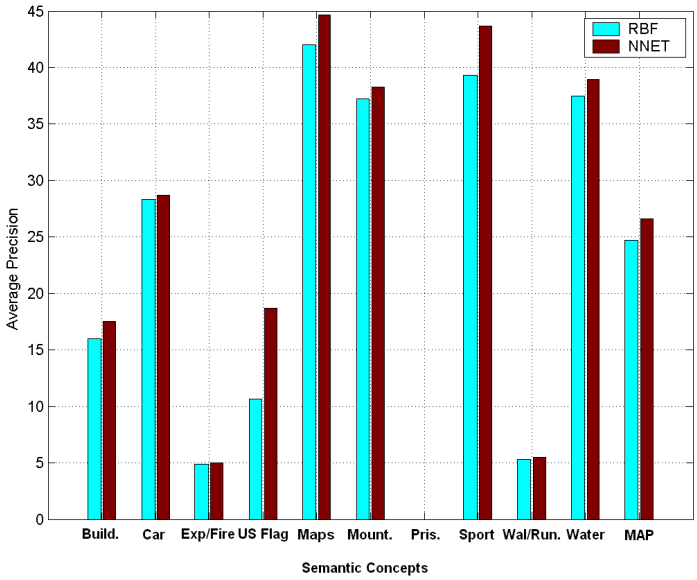


FIG. 3.16 – Performance du réseau RBF et NNET.

tissage plus faible. Ces résultats sont prévisibles puisque la règle de décision de ce dernier prend en compte seulement la probabilité *à posteriori*, alors que le NNET, au contraire, convertis cette probabilité sous la forme de fonctions de masses m_i, qui sont alors combinés en utilisant la règle de Dempster-Shafer. Les sorties de la fusion peuvent être présentées comme des degrés de croyance pour chaque classe. Ainsi, on peut dire que cette approche permet au processus décisionnel d'avoir une option de rejet, qui améliore les performances de la fusion comparée aux autres méthodes dites probabilistes.

Deuxièmement, NNET présente de plus fortes améliorations de détection pour les concepts (US FLAG, MAPS, SPORTS) que sur le reste. On peut expliquer cela par le grand nombre de fausse décision dans la classification, utilisant juste la probabilité *à posteriori*. La théorie des évidences atténue cet inconvénient, en présentant un degré de croyance dans le concept, avec un taux d'ignorance du système. Par ailleurs, NNET s'exécute en moins de 2 minutes par concept, ce qui revient à moins 1H20 pour les 36 concepts.

Pour une meilleure évaluation des performances de la fusion par NNET sur le schéma du CBVR, d'autres mesures peuvent être ajoutées comme : F-mesure (F-meas), le taux de classification des positifs (CR^+) et l'erreur (BER) "Balanced Error Rate". La Table 3.4 montre l'efficacité de la méthode de fusion par NNET, avec un gain au niveau du MAP, F-meas, CR^+ et une baisse du taux d'erreur "BER", par comparaison au réseau RBF et au système unimodal par SVM.

De façon globale, cette expérimentation a mis en exergue l'importance de la fusion dans notre système d'indexation et de recherche, avec un gain en précision moyenne de 22.06% par rapport au meilleur score de classification unimodal obtenu par le descripteur EDH.

	Classification (%)	Fusion (%)	
Technique	SVM (EDH)	Réseau RBF	NNET
MAP	21.39	24.85	26.11
F-meas	5.01	8.99	11.64
CR^+	2.71	5.08	6.77
BER	48.65	47.46	46.63

TAB. 3.4 – Comparaison des performances entre les résultats de la classification unimodal par SVM et de la fusion de classifieurs via le réseau RBF et NNET.

3.3.2 Evaluation de la méthode PENN

Dans le but d'évaluer la méthode PENN [9], nous avons besoin cette fois-ci d'une base de données avec plus de concepts annotés et d'extraire différents descripteurs pour une meilleure analyse de la relation descripteur/concept et de voir l'importance de la fusion post-classification dans notre système d'indexation. Pour cela, nous allons utiliser les données de TRECVid 2007 déjà décrites dans la section 1.4.1.2. Nous appliquerons le même processus que celui utilisé sur TRECVid 2005 : (1) Extraction des descripteurs (on utilisera ici un ensemble de cinq types de descripteurs MPEG-7 visuels globaux basés sur la couleur, texture, forme, mouvement et le détecteur de face, comme suit [KSp] : ScalableColor (SCD), ColorLayout (CLD), ColorStructure (CSD), ColorMoment (CMD), EdgeHistogram (EHD), HomogeneousTexture (HTD), StatisticalTexture (STD), Contour-based Shape (C-SD), CameraMotion (CM), MotionActivity (MAD), FaceDetector (FD)), ensuite (2) la classification par les SVMs (i.e. une SVM binaire par descripteur) et enfin (3) la fusion de classifieurs par le PENN [BH08b].

La plupart des descripteurs MPEG-7 ont été déjà présentés dans la section 2.1.2. Dans cette expérimentation et dans le cadre du projet NoE K-Space, trois autres descripteurs ont été ajoutés :

- **Color Moment Descriptor (CMD)** fournit des informations différentes sur la couleur de celle donnée par les autres descripteurs de couleur. Il est obtenu par le calcul de la moyenne et la variance de chaque couche de l'espace de couleur LUV d'une image ou d'une région [Ada07].
- **Statistical Texture Descriptor (STD)** est basé sur des statistiques à partir de la matrice de cooccurrence, telles que : l'énergie, la probabilité maximale, le contraste, l'entropie, etc [Ada07], pour modéliser les relations entre les pixels dans une région d'une configuration de niveau de gris dans la texture. Cette configuration varie rapidement avec la distance pour les textures fines, et lentement pour les textures grossières.
- **Face Descriptor (FD)** détecte et localise les parties frontales des visages dans les images-clés des plans vidéos, en présentant quelques statistiques (e.g. le nombre de visages, la taille de la plus grande face), en utilisant la méthode de détection de visage mis en oeuvre dans OpenCV [Ope], fourni par l'institut JRS.

[9] Perplexity-based Evidential Neural Network

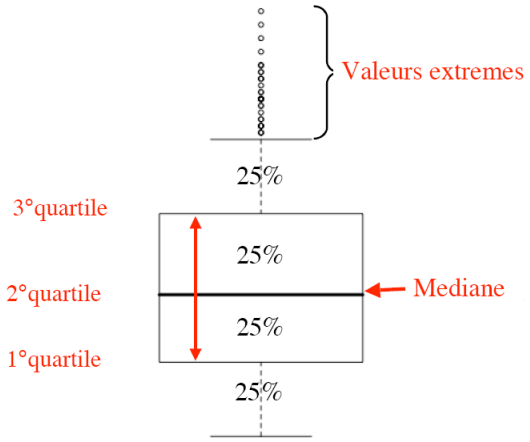


FIG. 3.17 – Variations autour d'une boîte à moustache (Boxplot).

La pertinence des différents descripteurs par rapport aux concepts de haut-niveau peut être obtenue grâce à l'étude de la distribution de la perplexité. De ce fait, le Boxplot (Fig. 3.17) fournit un excellent résumé visuel de nombreux aspects importants d'une distribution. Les lignes inférieures et supérieures expriment l'intervalle de variation des données. La partie inférieure et supérieure de la boîte indiquent les $Q_1 : 1^{er}$ et $Q_3 : 3^{ème}$ quartile. La ligne dans la boîte indique la valeur médiane des données.

La Fig. 3.18 montre la perplexité normalisée pour chaque descripteur avec son concept qui présente la plus forte relation. En effet, on peut dire que le SCD est plus adapté pour le concept SKY " 14", EDH pour ROAD " 13", etc. La première observation porte sur l'obtention de la même valeur médiane de la perplexité pour SCD, CLD, CMD, CSD, où la couleur est plus discriminante. Deuxièmement, C-SD donne le plus petit Q_1 de la perplexité normalisée pour toutes les données, suivie par EDH et SCD. En troisième lieu, il semble que EHD est très utile dans la détection des concepts avec de fortes représentations de contours comme pour SPORTS et ROAD. De même pour C-SD. En revanche, MAD présente un grand intervalle de perplexité, mais donne de faibles valeurs, pour les concepts WALKING/RUNNING, PEOPLE MARCHING où l'activité du mouvement est existante, forte et peut être détectée facilement. Enfin, FD est un descripteur efficace pour détecter FACE et PERSON, chose plutôt attendue.

La Fig. 3.19 présente la variation de la pondération normalisée *vs* les descripteurs visuels. On remarque que l'importance de chaque descripteur varie selon le concept concerné. Les descripteurs de couleurs et de textures ont plus de poids pour les concepts VEGETATION, BUILDING, SKY, etc". les descripteurs CM et MAD sont plus sensible à WALKING/RUNNING et PEOPLE MARCHING. Enfin, FD présente une grande sensibilité essentiellement pour les plans vidéo présentant des humains et particulièrement FACE.

Maintenant, nous souhaitons étudier le comportement des descripteurs par rapport aux concepts et sur les performances du système ? La Fig. 3.20 compare les performances d'un système sans pondération que nous appellerons dans la figure "No-Weight" (i.e. tous les

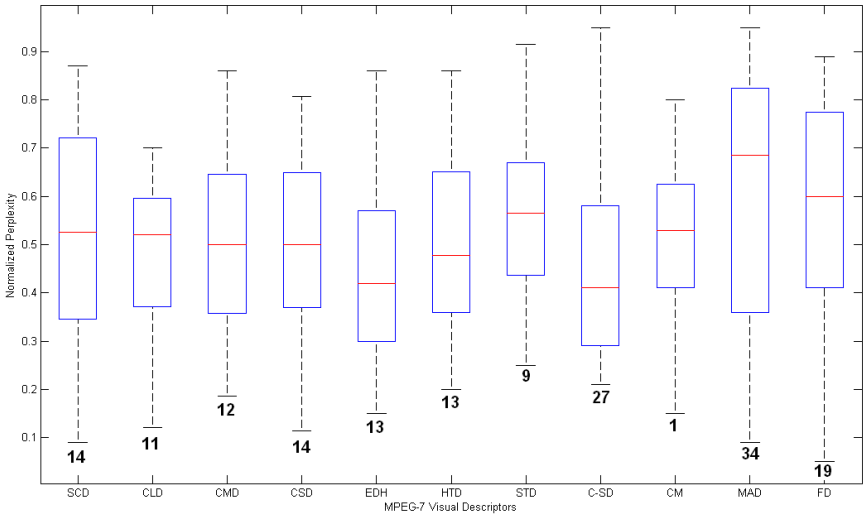


FIG. 3.18 – Boxplot représentant la variation de la perplexité normalisée des descripteurs visuels.

descripteurs sont considérés avec la même importance dans la combinaison pour tous les concepts sémantiques) avec quatre modèles de fonctions de pondérations différentes : Softmax, Sigmoïde, Gaussienne et Verhulst. Notre modèle basé sur la fonction d'évolution de Verhulst présente les meilleures performances sur la précision moyenne pour plusieurs concepts. PENN donne plus d'importance à FaceDetector, ContourShape, ColorLayout, ScalableColor, EdgeHistogram que pour les autres descripteurs. Concernant le concept FACE, l'amélioration atteint 11%, ce qui la met en première position en terme d'amélioration.

Les autres modèles produisent des résultats respectables et comparables en générale, avec quelques diminutions dues aux nombreuses situations de conflits et les limitations des données. Ceci, expliquent aussi les cas extrêmes obtenus pour les concepts COURT, PRISONER, US FLAG, AIRPLANE, EXPLOSION/FIRE.

Comme pour l'évaluation de la partie précédente, l'efficacité et la qualité de notre système sont directement résumées dans la Table 3.5. Les résultats sont calculés sur tous les concepts et sur les 10 concepts les plus représentés dans la base qu'on notera par (MAP@10, F-meas@10, CR^+@10, BER@10). Le PENN "Verhulst" permet une augmentation significative de la détection obtenue par l'amélioration du MAP, F-meas, CR^+, et une baisse remarquable de l'erreur globale BER comparant à un système simple de fusion à base du NNET "No-Weight". Ce mode offre la possibilité de retrouver en premier lieu les plans visuellement et sémantiquement similaires.

En résumé, nos deux propositions NNET et PENN ont montré que l'utilisation de la fusion à un haut-niveau de la description par l'adaptation de la théorie des évidences ainsi que l'intégration des relations entre descripteurs et concepts permettent d'améliorer de façon sensible le système d'indexation et de recherche de plans vidéo. Dans ce qui va suivre, nous

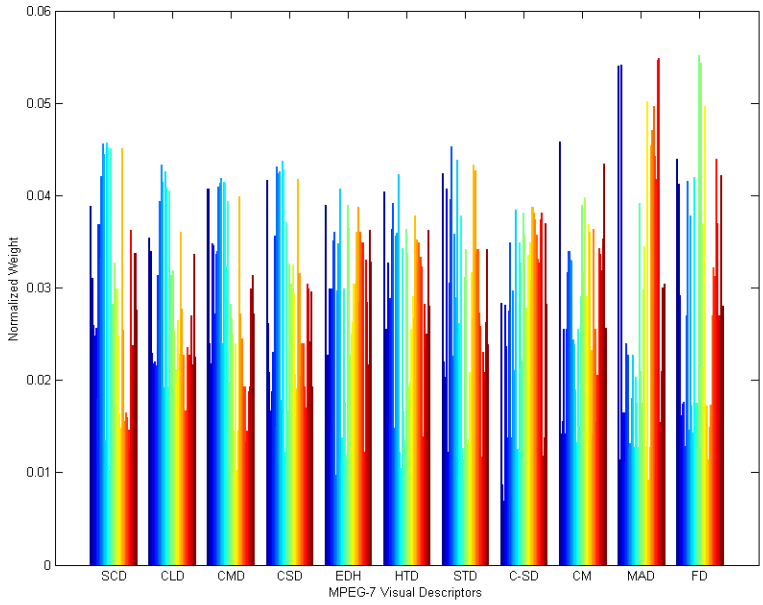


FIG. 3.19 – Comportement des descripteurs face aux concepts. SCD : ScalableColor, CLD : ColorLayout, CMD : ColorMoment, CSD : ColorStructure, EDH : EdgeHistogram, HTD : HomogeneousTexture, STD : StatisticalTexture, C-SD : Contour-based Shape, CM : CameraMotion, MAD : MotionActivity, FD : FaceDetector.

TAB. 3.5 – Comparaison de performances.

Méthodes / Évaluation	NNET"No-Weight" (%)	PENN"Verhulst" (%)
MAP	12.69	13.29
MAP@10	33.70	35.30
F-meas	11.84	14.10
F-meas@10	38.75	40.79
CR^+	11.93	13.43
CR^+@10	40.69	41.74
BER	45.02	44.13
BER@10	38.00	36.52

nous intéresserons à l'étude de l'apport et du comportement de la fusion de descripteurs bas-niveau dans notre système.

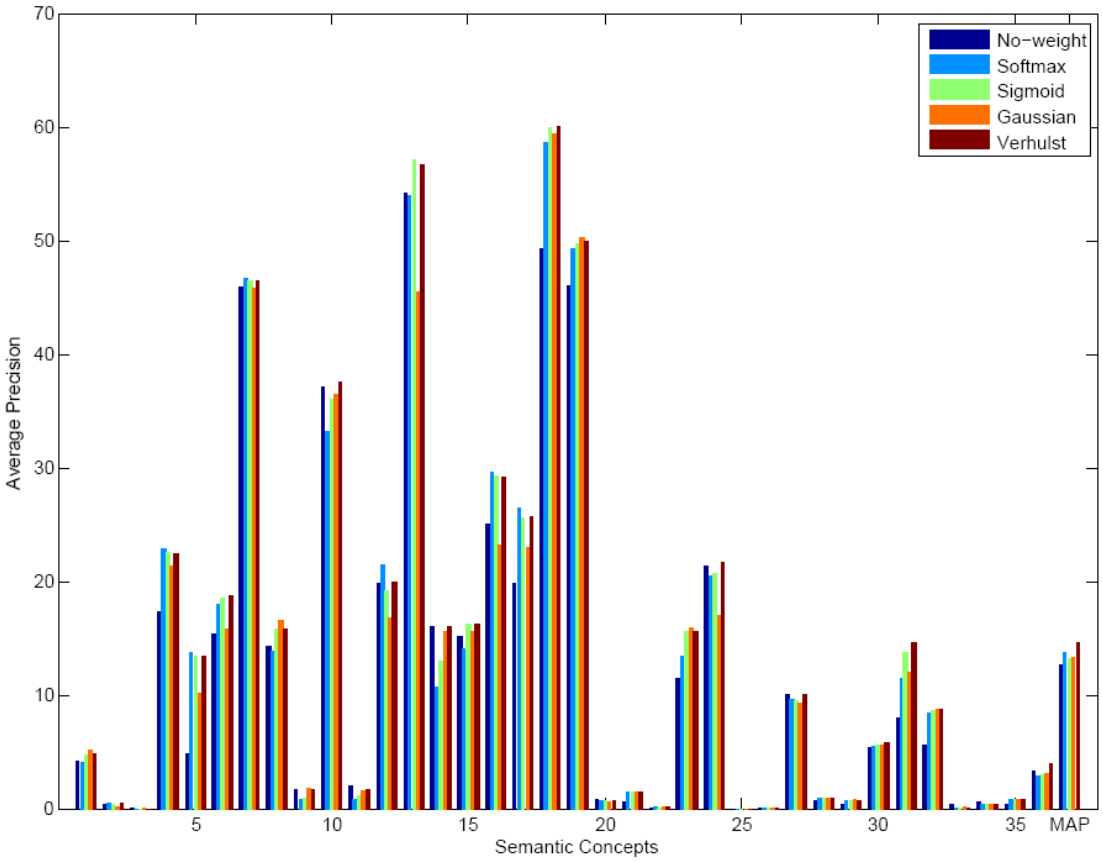


FIG. 3.20 – Comparaison de performances des 5 approches sur les 36 concepts. Le PENN basé sur le modèle de Verhulst surclasse les autres approches par une combinaison pondérée générique.

3.4 Fusion de descripteurs bas-niveau

La fusion de descripteurs bas-niveau (*Low-Level Feature LLF*) appelée aussi fusion de descripteurs / signatures prend tout (i.e. les descripteurs) en compte, les considérant comme compétitifs et complémentaires. Elle construit un nouveau vecteur en utilisant une des deux approches suivantes : statique (e.g. concaténation, par l'utilisation d'opérateurs simples) ou dynamique (e.g. réduction de dimensionnalité, théorie des probabilités, etc).

Par ailleurs, la reconnaissance des formes a pour objectif la classification des données d'entrée dans une des M classes proposées. Comme le montre la Fig. 3.21, il y a deux composantes : l'analyse des descripteurs et la classification. L'analyse des descripteurs est effectuée en deux étapes : l'extraction des descripteurs suivie par la transformation des descripteurs. Dans l'étape d'extraction des descripteurs, les informations hétérogènes (e.g. la couleur, la texture, le mouvement, l'audio, le texte) de l'objet sont extraites comme sous la forme d'un vecteur x_i de p_i-dimensions. Dans l'étape de transformation de descripteurs, le vecteur x_i est transformé en un vecteur caractéristique f_i de dimension m ($m \leq p$).

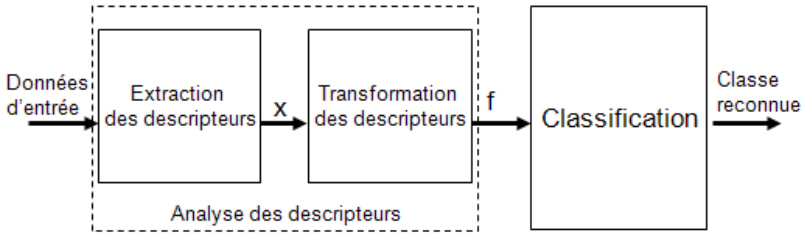


FIG. 3.21 – Système de classification.

Dans cette section, nous proposons un panorama des principales méthodes de transformation de descripteurs basées sur deux approches : (1) la fusion et (2) la sélection des descripteurs bas-niveau, pour améliorer les performances de l'apprentissage dans la prochaine étape "la classification". Tout d'abord, nous aborderons un point essentiel qui est la normalisation des données. Ensuite, nous détaillerons les deux approches citées ci-dessus.

La normalisation des données

Les descripteurs bas-niveau peuvent avoir des échelles de valeurs différentes. On citera par exemple une composante de couleur est dans l'intervalle [0,255], alors qu'une composante de texture à base de l'énergie de Gabor peut être dans l'intervalle [-10,10]. Pour cela, une étape de normalisation est nécessaire. Elle aura la tache de transformer chaque composante du vecteur dans un intervalle de valeur commune. Cette étape est plus importante en particulier lorsque l'algorithme d'apprentissage utilise une fonction de distance (e.g. KNN, SVM à noyaux, RBF,...), afin qu'aucune composante n'influence plus que d'autres l'apprentissage [AQ07]. Cependant, l'utilisation directe de ces variables donnera de façon implicite plus de poids aux variables de plus forte dispersion, annihilant presque complètement l'effet des autres variables.

Il existe dans la littérature plusieurs méthodes de normalisation [Dau]. La méthode la plus utilisée consiste en une transformation linéaire des données dans un intervalle [a,b] (Equ. 3.60) :

$$x_n = b\frac{x - min}{max - min} + a \tag{3.60}$$

où x est une composante d'un descripteur, min et max sont respectivement le minimum et le maximum de la composante, sur tout l'ensemble de l'apprentissage.

3.4.1 Concaténation et utilisation de simples opérateurs

La première stratégie est basée sur l'opérateur de concaténation. Les descripteurs bas-niveau sont combinés dans un vecteur unique appelé *merged fusion* (D_{merged}) (Eq. 3.61).

$$D_{merged} = [DCD|CLD|SCD|...|PMD] \tag{3.61}$$

Tous les descripteurs doivent avoir la même échelle de dimension. Cette méthode présente de bonnes performances pour la classification d'images, comme le démontre l'étude de Spyrou et al. [SBM+05]. D'autres opérateurs peuvent être utilisés, comme la moyenne des sorties des divers modules de traitement ou caractéristiques. Elle ne requiert aucune compilation de données, ces dernières subissent qu'une simple normalisation avant d'être sommées. Il est intéressant de réaliser une pondération qui exprimera la confiance accordée à chacune des caractéristiques.

Cette méthode a été utilisée pour la fusion de descripteurs dans les travaux de Rautiainen et al. [RS05] sur TRECVid 2003, où on observe l'apport de l'opérateur de fusion par la moyenne comparant à d'autres opérateurs comme le *Min* et le *Max*.

3.4.2 Réduction de dimensionnalité

La seconde stratégie a pour objectif la création d'un petit nombre de variables qui décrivent aussi bien les individus de la base initiale, habituellement en grand nombre. Ces nouvelles variables seront moins redondantes que les variables initiales. La réduction de la dimensionnalité est souvent difficile et risquée, mais présente beaucoup d'enseignements car elle amène à une confrontation d'opinions diverses quant à l'importance contestée de certaines variables.

Plusieurs travaux [CM00, WMS00] ont été menés et présentés comme des solutions aux problèmes dus aux espaces de grandes dimensions. Un état de l'art complet incluant les méthodes linéaires et non-linéaires a été dressé dans [Ber04].

Mathématiquement, le problème de réduction de la dimension peut être formuler comme : *étant donnée une variable aléatoire* $X = (X_1, ..., X_n)$ *de dimension initiale* n*, il s'agit de trouver une représentation* $U = (U_1, ..., U_p)$ *de dimension réduite* p *avec* $p < n$*, qui exprime selon un certain critère, la même information initiale.*

Les techniques linéaires reposent sur le même principe. Elles combinent les données originales de façon linéaire. La différence est dans la manière de déterminer les matrices de transformation (W et A), c'est à dire :

$$u_i = w_{1,i}x_1 + ... + w_{n,i}x_n \quad \text{pour} \quad i = 1, ..., p \tag{3.62}$$

$$\text{ou bien} \quad U = WX \tag{3.63}$$

avec W est une matrice $p \times d$ de transformation linéaire. La transformée inverse est :

$$X = AU \tag{3.64}$$

avec A la matrice $d \times p$ de transformation inverse.

Les techniques non-linéaires sont souvent itératives et obtenues soit en utilisant des projections non-linéaires, soit en minimisant une fonction de coût liée aux distances inter-vecteurs. Le but est de garder les mêmes distances entre les vecteurs dans l'espace original et leurs correspondances dans l'espace transformé. Autrement dit, l'objectif est de transformer s vecteurs initialement de dimension n dans un nouvel espace de dimension p avec $p < n$, tel que :

$$\forall \;\; 1 \leq i, j \leq s, \quad d(x_i, x_j) = \delta(\hat{x_i}, \hat{x_j}) \tag{3.65}$$

où d est une mesure de similarité entre les vecteurs dans l'espace original, δ est la fonction de distance entre vecteurs dans l'espace transformé. $\hat{x_i}$ et $\hat{x_j}$ sont respectivement les transformées de x_i et x_j dans l'espace transformé [Ber04].

Dans cette thèse, nous allons nous intéresser aux méthodes de réduction de dimensions les plus citées dans le domaine de l'indexation, en particulier à des approches statistiques sophistiquées, essentiellement de l'analyse linéaire (ACP, LDA, NMF), ainsi qu'une technique non-linéaire à partir de réseau de neurones, permettant une réduction maximale du nombre de variables avec une perte minimale d'informations.

Avant cela, il est important de répondre à la question suivante : *Pourquoi réduire la dimension d'une base de données, alors que ce processus entraîne généralement une perte d'information ?*

- La première raison qui est la plus évidente, mais aussi la moins importante, est celle de réduire la quantité d'informations que les algorithmes auront à traiter, réduisant ainsi, le temps de calcul, l'encombrement de la mémoire et le stockage de l'information.
- Réduire le nombre de variables à 2 permet une représentation visuelle plane des données, ainsi que l'utilisation du meilleur système d'analyse : *l'oeil* et son fantastique système de détection de regroupements, d'alignements, etc.
- Mais la raison la plus importante est celle de la crédibilité de notre modèle. Un modèle ne prenant en entrée que peu de variables sera plus crédible qu'un modèle utilisant un grand nombre de variables d'entrée. Ce dernier est souvent considéré comme peu intuitif avec pour conséquence la prolifération de modèles déclarés «excellents», mais en réalité inutilisables. Son application sur des nouvelles données produit des résultats de mauvaise qualité, en raison d'un trop grand nombre de variables prises en compte

par le modèle (mauvaise généralisation).

Par conséquent, la réduction de dimensionnalité est un exercice à la fois indispensable et difficile. L'analyste se doit de lui consacrer le temps nécessaire sous peine de construire des modèles de bonne qualité pour certains cas particuliers, mais qui ne peuvent être généralisés.

3.4.2.1 Analyse en Composantes Principales (ACP)

L'ACP est la principale technique linéaire de réduction de dimensionnalité [DK82]. Cette analyse non-supervisée a pour objectif de décrire un ensemble de données par de nouvelles variables en nombre réduit [10].

Bien que l'objectif est d'utiliser qu'un petit nombre de *CP Composantes Principales (CP)*[11], l'ACP en construit initialement p, autant que de variables initiales et classées par ordre décroissant. Saporta [Sap90] a montré que la recherche des axes (CP) du sous-espace peut se faire de manière séquentielle. Il commence par l'axe qui décrit le mieux les données, puis le deuxième qui en plus doit être orthogonal au premier, et ainsi de suite [Tol06].

Ensuite, l'analyse décidera du nombre k de *CP* à retenir et qui représentent au mieux la base de données X après projection (i.e. remplacer les observations initiales par leurs projections orthogonales dans le sous-espace à k dimensions défini par les k premières *CP*). Cela revient à rendre J minimal (Equ 3.66), de sorte que le nuage de points projetés soit le moins déformé possible, autrement dit, la variance du nuage de points projetés est maximale.

$$J = \frac{1}{2}\sum_{i=1}^{N} ||X_i - \text{proj}(X_i)||^2 \tag{3.66}$$

Plus généralement, le problème de minimisation de J revient à calculer les k vecteurs propres de la matrice de covariance S (Eq. 3.67) associée aux plus grandes valeurs propres.

$$S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T \tag{3.67}$$

où, $\bar{x}$ est la moyenne sur N échantillons dans l'ensemble d'entraînement, x_i est le $i^{\text{ème}}$ échantillon.

Le calcul de *CP* peut être optimiser par la méthode de Décomposition en Valeurs Singulières *SVD*[12], permettant de représenter la matrice S de taille (NxM) sous la forme suivante :

$$S = UDV^t \tag{3.68}$$

avec, $UU^t = U^tU = I_N, V^tV = I_N$ et $D = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N)$.

[10] Autre que la réduction de la dimension, l'ACP est aussi utilisée pour filtrer le bruit, compresser les données, etc.

[11] Les Composantes Principales sont des variables qui s'avèrent être deux à deux décorrélées.

[12] SVD est un algorithme qui permet d'exprimer une matrice comme le produit de trois matrices particulières.

Il n' y a pas de méthode directe pour sélectionner automatiquement le nombre k de *CP*, il est décidé soit *à priori*, soit par seuillage des valeurs propres (e.g. critère de Kaiser), ou enfin par les expérimentations. Par ailleurs, la variance dépend de l'ordre de grandeurs des descripteurs, lorsque ces derniers sont de types différents, les données doivent être centrées réduites. La matrice de covariance devient une matrice de corrélation.

Cependant, l'ACP est un simple changement de repère, son recours à l'algèbre linéaire comme outil mathématique principal et sa simple interprétation géométrique représentent sa grande force. Mais elle est aussi sa faiblesse. En effet, rien ne dit que des nouvelles variables plus complexes que celles résultantes d'un changement de repère ne permettraient pas une description plus économe de données. L'ACP a donc reçu de nombreuses généralisations, basées sur des transformations non-linéaires des variables originales. Le lecteur intéressé pourra trouver des informations sur les noyaux-ACP dans [SSM98].

Alors qu'une ACP considère seulement les moments de second ordre pour décorréler [13] les dimensions. L'Analyse en Composantes Indépendantes ACI [GCA98] considère des moments d'ordre supérieur afin de rendre les composantes indépendantes [14]. Cependant, une ACI ne permet pas de réduire le nombre de dimensions, mais de les rendre indépendantes. En complément avec une ACP, elle permet d'obtenir des composantes indépendantes dans un espace de dimensions réduit.

3.4.2.2 Analyse Discriminante Linéaire (LDA)

Contrairement à l'ACP et l'ACI qui sont des techniques non-supervisées, la LDA réduit la dimension tout en préservant au maximum les classes [Fak90]. Cette méthode est utile lorsque les fréquences inter-classes sont inégales [Tol06].

Le point de départ de la LDA est une matrice X de N données observées (nuage de points) dont les éléments sont identifiés par les k classes possibles, de centre de gravité g et de matrice variance-covariance V. Ce nuage est partagé en q sous-nuages par la variable "classe". Chaque sous-nuage (classe w_k) d'effectif n_k est caractérisé par son centre de gravité g_k et sa matrice variance-covariance V_k.

En utilisant la relation de Huyghens qui stipule que la matrice de covariance V (Eq. 3.69) estimée à partir des observations, peut être décomposée en deux matrices différentes, l'une B (*Between*) matrice de variance inter-classe, qui rend compte de la dispersion des centroïdes des classes g_k autour du centre global g. Et l'autre W (*Within*) matrice de variance intra-classe, qui est la moyenne des k matrices variance-covariance des classes V_k.

$$\begin{cases} V = B + W \\ B = \frac{1}{N} \sum_{k=1}^{q} n_k (g_k - g)(g - g_k)^T \\ W = \frac{1}{N} \sum_{k=1}^{q} n_k V_k \end{cases} \tag{3.69}$$

On peut donc chercher un ensemble d'axes qui résume au mieux la variance intergroupes (i.e. qui disperse au maximum les observations si elles appartiennent à deux groupes différents) et qui, dans le même temps, minimise la variance intra-groupe (i.e. représentation très proches des observations d'un même groupe).

[13] La corrélation mesure l'existence d'une relation linéaire entre les variables.
[14] La dépendance mesure l'existence de n'importe quelle relation entre les variables.

Cela dit, les deux critères ne peuvent pas être satisfaits simultanément (i.e. pour une même direction de projection). Le critère de Fisher J (Eq. 3.70) réalise un compromis entre ces deux objectifs. De manière formelle, le problème est de trouver la matrice de projection A qui maximise J. Remarquons que ce critère suppose une répartition gaussienne des données de chaque classe, dans l'espace de projection.

$$J = \frac{A^T B A}{A^T V A} \tag{3.70}$$

Le but est de favoriser le regroupement des points d'une même classe et de maximiser la distance entre les clusters obtenus. En transformant A en un ensemble de vecteurs $[a_1, ..., a_n]$, le problème devient [DHS01] :

$$V^{-1} B a_i = \lambda a_i, \forall i \in \{1, ..., p\} \tag{3.71}$$

A est composé de vecteurs propres associés aux p premières valeurs propres triées par ordre décroissant de $V^{-1}B$, $p < k$ étant fixé par l'utilisateur.

3.4.2.3 Factorisation en Matrices Non-négatives (NMF)

La NMF est une méthode générale de décomposition matricielle [LS00]. Elle permet d'approximer toute matrice V de taille (NxM) et dont les éléments sont tous positifs, grâce à une décomposition de la forme $V \approx WH$, où $W \in \mathbb{R}^{N\text{x}k}$ et $H \in \mathbb{R}^{k\text{x}M}$. L'originalité de la NMF réside dans les contraintes de non-négativité qu'elle impose à W et H. Ces contraintes font que les vecteurs de base comportent beaucoup de 0 et que leurs parties non nulles se chevauchent rarement. La représentation d'un objet (décrit par un vecteur de réels positifs) comme une somme de ces vecteurs de base, correspond alors à l'intuition d'une décomposition par parties. Ici, la matrice codée H peut être considérée comme le nouveau vecteur de dimension réduite.

En comparaison avec la SVD, qui ne permet pas d'interpréter ni les vecteurs de base ni les nouvelles coordonnées réduites, à cause de la présence de coefficients négatifs.

Déterminer les matrices W et H revient à minimiser la distance entre la matrice initiale et le produit WH. Plus précisément, il faut diminuer la norme de Frobenius (Equ. 3.72) sous les contraintes de non-négativité.

$$min_{W,H} ||V - WH||^2 = \sum_{ij} (V_{ij} - (WH)_{ij})^2 \tag{3.72}$$

D'autres normes peuvent être utilisées, comme par exemple [XF07] :

$$E(V, WH) = \sum_{ij} (V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij}) \tag{3.73}$$

Ceci revient à un problème d'optimisation non trivial. Dans [LS00], l'auteur propose de le résoudre en initialisant W et H aléatoirement, puis en effectuant les mises à jour suivantes :

$$\begin{cases} H_{ij} \leftarrow H_{ij} \frac{(W^T V)_{ij}}{(W^T WH)_{ij}} \\ W_{ij} \leftarrow W_{ij} \frac{(VH^T)_{ij}}{(WHH^T)_{ij}} \end{cases} \tag{3.74}$$

Avantages et inconvénients

Toutes les techniques linéaires citées permettent de réduire la dimension. Cependant, si le nombre de dimensions est élevé [Tol06], il se peut que cette réduction ne soit pas suffisante sans perdre de l'information au point que le résultat ne serait plus exploitable. En effet, les problèmes liés à la *malédiction de la dimension* [15] se manifestent selon Weber et al. [WSB98] à partir d'une dimension de 16. Or, on trouve des vecteurs de couleurs de dimension 256.

De plus, ces techniques nécessitent une mise à jour de l'analyse en cas d'ajout de nouvelles données (i.e. recalcule de la matrice de covariance, axes principaux et enfin la projection des vecteurs dans le nouvel espace). La mise à jour est une opération extrêmement coûteuse lorsque le contenu de la base évolue régulièrement.

L'ACP est très sensible aux valeurs aberrantes, ainsi, il est préférable de détecter ces erreurs avant son application. S-A. Berrani [Ber04] a étudié ce problème à travers une expérience sur un ensemble de données réelles composé de 413412 vecteurs de dimension 24. Ces vecteurs sont des descripteurs locaux d'images couleurs. Une ACP a été réalisée sur cet ensemble de données. Les résultats montrent qu'une seule composante porte à elle seule l'essentiel de l'information avec un pourcentage de l'inertie expliquée égale à 96.04% (la valeur propre associée a cette composante est égale à 838.74, alors que la somme de toutes les valeurs propres associées aux 24 composantes est égale à 873.33). Un filtrage manuelle de 1004 vecteurs [16] a été effectué, suivi d'une ACP. Cette fois-ci, il faut retenir 19 composantes principales pour avoir un pourcentage de l'inertie égale à 96.04%.

Par ailleurs, l'ACP recherche les axes avec la plus grande variance ou d'indépendances pour l'ACI, permettant de supprimer les redondances. Martinez et al. [MK01] ont montré qu'elle est moins sensible si l'ensemble d'apprentissage est petit et donne de meilleurs résultats que la LDA. Cette dernière cherche les axes les plus discriminants avec l'hypothèse de gaussianité et nécessite un long calcul. Cependant, l'ACP ne préserve pas les classes. Une méthode a été proposée par Belhumeur et al. [BHK97], qui combine les deux techniques (ACP plus LDA) afin d'en profiter des caractéristiques de chacune.

3.4.2.4 Neural Network Coder (NNC)

Dans le même souci de réduction de la dimensionnalité, nous proposons ici, une méthode non-linéaire de fusion multimodale basée sur les perceptrons multi-couches. A travers la littérature, les MLPs ont été utilisés et évalués dans plusieurs travaux, en particulier ceux de Hinton et Salakhutdinov [HS06] qui construisent un auto-encodeur par une série de réseaux

[15] La malédiction de la dimension "dimensionality curse" fait référence aux difficultés de la gestion et du traitement des données dans les espaces de grande dimension. En statistique, elle exprime la relation entre la taille de l'échantillon de données et la précision de l'estimation. Donoho [Doh00] montre qu'il faut augmenter exponentiellement le nombre d'échantillons avec l'accroissement de la dimension pour garder le même niveau de précision. Les travaux de Beyer et al. [BGRS99] montrent aussi que plus la dimension augmente, plus les vecteurs ont tendance à devenir équidistants dans une distribution uniforme, ce qui engendre une instabilité des résultats. Le lecteur intéressé par plus de détails pourra se référer à l'état de l'art dans [Ber04].

[16] Les 1004 vecteurs possèdent des composantes aberrantes obtenues suites à des divisions proches de 0.

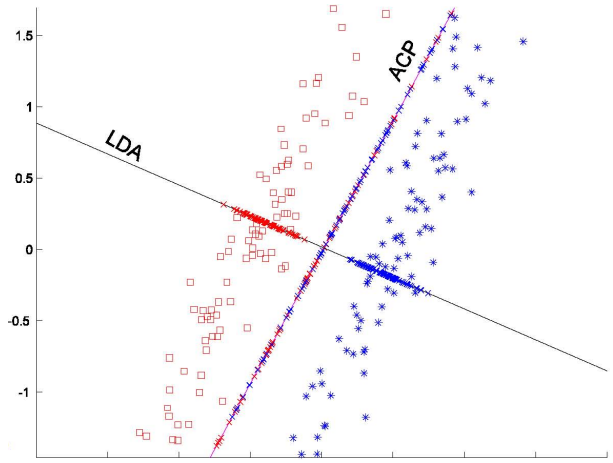


FIG. 3.22 – Exemple de réduction de dimensions par ACP et par LDA. Les deux méthodes permettent de passer de 2 à 1 dimension. L'axe fourni par la LDA sépare les données en classes, tandis que l'ACP les confond (Figure tirée de la thèse de S. Tollari [Tol06].)

de neurones entrainés par les machines de Bolzman et appliqués pour la reconnaissance de visages et des chiffres. Le décodage s'effectue par l'inversion du réseau, obtenant ainsi plusieurs couches cachées. Dans cette partie, l'originalité provient de l'adaptation innovatrice de l'outil à l'application et à la formalisation de l'information, qu'elle soit à traiter ou qu'elle soit celle résultante. D'autres techniques non-linéaires existent. Nous allons simplement citer les plus connues : Analyse en composantes curvilinéaires (ACC) [DH] basée sur un réseau de neurone SOM (*self-organized maps* - carte auto adaptative), reprise récemment dans les travaux de G. Lefebvre [Lef07], la technique *FastMap* [FL95].

Les perceptrons multi-couches [DN93] sont souvent utilisés en classification, en identification et en reconnaissance de formes. En fusion de données, ils permettent de traiter la variabilité de procédés complexes et la disparité des informations provenant des sources. En général, ils nécessitent des temps de traitement très courts. En fonction des valeurs d'entrées, ils évaluent des grandeurs de sortie. L'intérêt des réseaux de neurones, c'est qu'ils ne nécessitent pas de modèle formel du phénomène observé puisque leur fonctionnement est basé sur l'utilisation d'une connaissance obtenue par l'apprentissage.

De plus, ils sont constitués de neurones connectés entre eux de différentes manières [HJ94]. Le réseau est défini par :

- sa topologie ;
- la fonction d'activation des neurones (linéaire, sigmoïde,...) ;
- les méthodes d'apprentissage supervisées ou non.

La méthode d'apprentissage la plus connue est la rétro-propagation du gradient de l'erreur. Elle est utilisée pour les réseaux ayant une couche d'entrée, une couche de sortie et au moins une couche cachée. Le principe de la rétro-propagation consiste à présenter au réseau un vecteur d'entrée et de procéder au calcul de la sortie par propagation à travers

les couches. Ce résultat est comparé à la sortie souhaitée. On calcule ensuite le gradient de l'erreur obtenu qui est propagé de la couche de sortie à la couche d'entrée, afin de modifier le poids des entrées de chacun des neurones.

L'utilisation d'un MLP en tant qu'agent de fusion est particulièrement séduisante, grâce à sa capacité d'apprentissage, le réseau devrait pouvoir gérer au mieux les conflits qui apparaissent entre les divers agents de traitement. Dans [SBM+05], un MLP d'une couche cachée a été utilisé pour la fusion des descripteurs MPEG-7, cela a permis d'améliorer le taux de classification des images.

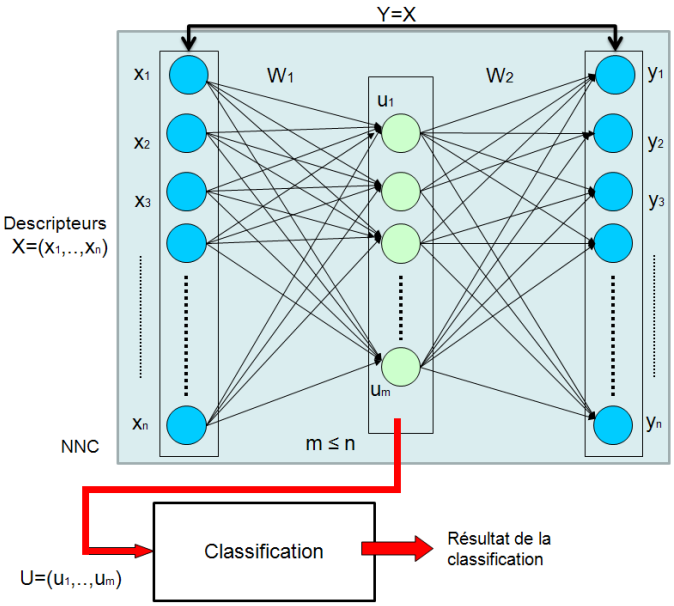


FIG. 3.23 – Représentation du codeur NNC.

Par ailleurs, plusieurs causes expliquent la dégradation des performances des techniques d'indexation [Ber04]. D'abord, ces techniques reposent généralement sur des hypothèses non-vérifiées quant à la distribution des descripteurs (e.g. distribution uniforme), et sont souvent validées en utilisant peu de descripteurs, souvent basiques. Ensuite, la minimisation du problème de la grande dimension par l'utilisation simple des méthodes d'analyse de données (e.g. ACP). Or, ces derniers entraînent souvent une baisse significative de la qualité des résultats. Pour cela, nous proposons l'utilisation d'une méthode à base d'apprentissage de données et du réseau multi-couches (Fig. 3.23). Elle présente trois couches (une couche d'entrée, une couche cachée et une couche de sortie) avec une clause particulière $Y = X$ (i.e. la couche de sortie est égale à la couche d'entrée), ce qui lui donne un aspect symétrique. La sortie de la couche cachée U va être prise comme le résultat du codage, qui sera introduit dans le système de classification.

Deux questions peuvent cependant être posées concernant le nombre de noeuds dans la couche cachée, qui est aussi la dimension (m) (moins il y en a, plus le modèle est robuste ;

plus il y en a, plus il est performant) et la nature des fonctions d'activation (Dans [Smi], il est indiqué que pour un perceptron, la fonction d'activation doit être la même dans tous les neurones de toutes les couches, alors que le réseau RBF *Radial Basis Function* présente une gaussienne dans la seule couche cachée et une fonction linéaire en sortie. Cependant, cette vision est considérée comme réductrice, pour cela, plusieurs fonctions d'activation sont présentées en couche de sortie). La réponse sera donnée dans la partie évaluation de la section 3.5, car elle dépend de la nature des données (X) et des descripteurs utilisés pour les représenter.

La réduction de la dimension permet en particulier de mettre un grand nombre d'informations entrantes dans le modèle sans se soucier de la redondance des informations ou de leurs inutilités. La couche cachée est un goulet d'étranglement dont seule l'information pertinente ressortira, combinée et transformée de manière non linéaire, pour faire coïncider les variations des données en entrée avec les variations de la quantité à prédire. Ainsi, on peut voir l'action des neurones cachés comme une division de l'information entrante.

Le passage par un codage permet d'avoir une nouvelle description de l'information plus compacte, riche et plus représentative. Dans notre étude, un programme se charge d'essayer toutes les architectures en faisant varier le nombre de neurones cachés et les fonctions d'activation utilisées, pour en retenir que la plus performante sur un jeu de test.

3.4.3 Fusion dans la théorie des Probabilités

Dans un autre registre de la fusion dynamique de descripteurs, nous allons nous intéresser à la théorie des probabilités [Sap90], traditionnellement utilisée pour modéliser l'incertitude dans de nombreuses disciplines et repose sur des bases théoriques solides. La plupart des techniques existantes utilisent l'approche Bayésienne pour opérer à la fusion. Cette approche permet d'incorporer l'incertitude dans la modélisation.

La décision est alors prise sur la base des données multi-sources et des connaissances à priori sur les états du système étudié. Dans la plupart des cas, on utilise comme règle de décision le maximum de vraisemblance ou encore le maximum de probabilité à posteriori (MAP).

Si l'on note $X = (x_1, ..., x_n)$ les observations fournies par une source et Ω l'ensemble des valeurs possibles de la caractéristique du système étudié que l'on cherche à estimer (état du système, position, orientation d'un objet, classe d'objets,...). Pour chaque $w \in \Omega$, on définit :

- la fonction de vraisemblance $p(X|w)$ par la probabilité d'avoir comme vecteur d'observation X sachant que la caractéristique du système vaut ω ;
- $p(w/X)$ la probabilité pour que la caractéristique soit w lorsque l'on observe X. D'après la formule de Bayes, on a :

$$p(w/X) = \frac{p(X/w)p(w)}{p(X)} \tag{3.75}$$

- $p(X)$ et $p(w)$ sont les probabilités que la source observe X et que la caractéristique du système ait pour valeur w respectivement.

Si on suppose que l'ensemble Ω est de cardinalité M *(fini)* (i.e. $\Omega = \{w_1, w_2, ..., w_M\}$), la formule de Bayes devient pour $i = \{1, ..., M\}$:

$$p(w_i/X) = \frac{p(X/w_i)p(w_i)}{\sum_{l=1}^{M} p(X/w_l)p(w_l)} \tag{3.76}$$

Il est clair que la fusion n'apparaît pas dans cette formule, mais intervient plutôt en amont pour calculer $p(X/w)$ à partir des fonctions de vraisemblances associées à chaque source s, $p(X^s/w); s = \{1, ..., L\}$ (L est le nombre de sources) et éventuellement d'autres informations sur la dépendance des sources et de leurs fiabilités respectives.

Si on suppose que les sources sont indépendantes de sorte que la fonction de vraisemblance $p(X/w)$ est donnée par :

$$p(X/w) = \prod_{s=1}^{L} p(X^s/w) \tag{3.77}$$

En utilisant la règle du *maximum de vraisemblance*, on choisira la valeur de w qui maximise la quantité précédente. Pour des raisons de simplifications, on utilisera *le logarithme* de la fonction de vraisemblance, appelé log-vraisemblance. Ceci revient donc à maximiser $\log[p(X/w)]$ ayant pour expression :

$$F(w) = \sum_{s=1}^{L} \log p(X^s/w) \tag{3.78}$$

Pour illustrer la mise en oeuvre de ce type de fusion, prenons le cas de L capteurs d'erreurs gaussiennes. Dans ce cas, la fonction de vraisemblance $p(X/w)$ est donnée par :

$$p(X^s/w) = |2\pi\Omega_s|^{-1/2} \exp\left\langle -\frac{1}{2}(X^s - w)^t \Omega_s^{-1}(X^s - w) \right\rangle \tag{3.79}$$

où $|\Omega_s|$ désigne le déterminant de la matrice variance-covariance de X^s. La fonction log-vraisemblance s'écrit alors [17] :

$$F(w) = \sum_{s=1}^{L} \left(-\frac{1}{2}\log((2\pi)^n |\Omega_s|) - \frac{1}{2}(X^s - w)^t \Omega_s^{-1}(X^s - w) \right) \tag{3.80}$$

L'estimation du maximum de vraisemblance de w est obtenue par la résolution des équations normales (dérivée partielle de F par rapport à w est nulle), on obtient :

$$\hat{w} = \frac{\sum_{s=1}^{L} \Omega_s^{-1} X^s}{\sum_{s=1}^{L} \Omega_s^{-1}} \tag{3.81}$$

Dans le cas où l'on dispose de deux capteurs ($L = 2$) et que chaque capteur fournit une mesure de (w, x_i), avec $s = \{1, 2\}$; d'après l'équation précédente, l'estimation du maximum de vraisemblance de w est donnée par (posons $\sigma_s^2 = \Omega_s^{-1}$) :

[17] Rappel : $\ln(\exp(t)) = t$

$$\hat{w} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2 \tag{3.82}$$

Ce dernier résultat souligne les deux types d'utilisation que l'on peut faire de l'approche Bayésienne en fusion de données :

- fusionner à un instant donné les informations issues de sources multiples (on parle de fusion de sources parallèles) ;
- fusionner les informations issues d'une même source mais à des instants différents.

3.4.4 Fusion dans la théorie des évidences

Comme décrit dans la section 3.2.2, la théorie des évidences manipule des sous-ensembles de l'espace de discernement 2^Ω plutôt que des singletons, ce qui lui donne une grande souplesse de modélisation pour de multiples situations de fusion. Ceci se traduit par d'autres représentations de l'information à travers : l'incertitude, l'imprécision et l'ignorance à l'aide de la fonction de masse m, de plausibilité pl et de crédibilité bel. Cette théorie permet de mesurer les conflits existants entre les sources et de les interpréter en termes de fiabilité des sources [Blo03].

La théorie des évidences permet aussi de combiner des fonctions de masses issues de différentes sources d'informations. Plusieurs modes de combinaisons ont été développés. Il existe principalement deux types de combinaison : conjonctive et disjonctive. Elles ont été déclinées en un grand nombre d'opérateurs de combinaison dont les combinaisons mixtes.

- *Combinaison conjonctive :* (ou règle orthogonale de Dempster-Shafer) permet de combiner deux fonctions de masses ou plus, en une seule, en considérant leurs intersections. Elle est donnée pour tout $A \in 2^\Omega$ par :

$$m(A) = (m_1 \oplus m_2 \oplus ... \oplus m_n) = \sum_{B_1 \bigcap ... \bigcap B_n = A} \prod_{j=1}^{n} m_j(B_j) \tag{3.83}$$

  Si nous supposons être en monde clos, il est important de forcer la masse de l'ensemble vide à 0 ce qui pousse à normaliser les autres masses. Nous obtenons ainsi, la forme normalisée de la fonction de masse combinée, pour tout $A \in 2^\Omega$.

$$\begin{cases} m(A) = \frac{1}{1-k} \sum_{B_1 \bigcap ... \bigcap B_n = A} \prod_{j=1}^{n} m_j(B_j) \\ m(\emptyset) = 0 \end{cases} \tag{3.84}$$

  où $k = \sum_{B_1 \bigcap ... \bigcap B_n = \emptyset} \prod_{j=1}^{n} m_j(B_j) \prec 1$ (k est une *mesure de conflit*). La combinaison obtenue a tendance à renforcer la croyance sur les décisions pour lesquelles les sources sont concordantes et à l'atténuer en cas de conflit.

- *Combinaison disjonctive :* elle est obtenue cette fois-ci en considérant les unions. La combinaison de n fonctions de masse m_j est donnée par :

$$m(A) = \sum_{B_1 \bigcup ... \bigcup B_n = A} \prod_{j=1}^{n} m_j(B_j) \tag{3.85}$$

Il est clair que $m(\emptyset) = 0$ par cette combinaison. Le conflit ne peut pas donc exister. En contrepartie, les éléments focaux de la fonction de masse résultante seront élargis, ce qui pousse à une perte de spécificité. Cette formalisation ne permet pas une prise de décision sur les singletons, sauf dans le cas où les sources sont en accord sur ces derniers. Cependant, la combinaison disjonctive est intéressante si nous ne savons pas modéliser les fiabilités des sources, leurs ambiguïtés et imprécisions.

– *Règle Smets :* aborde le monde ouvert (les solutions peuvent être autre que dans Ω), qui est la plus utilisée aujourd'hui. Smets propose la combinaison suivante [Sme94] :

$$\begin{cases} m(A) = \sum_{B_1 \bigcap \ldots \bigcap B_n = A} \prod_{j=1}^{n} m_j(B_j) \\ m(\emptyset) = \sum_{B_1 \bigcap \ldots \bigcap B_n \neq \emptyset} \prod_{j=1}^{n} m_j(B_j) \end{cases} \tag{3.86}$$

– *Combinaison mixte :* cherche à conserver les avantages de la combinaison conjonctive et disjonctive. Dubois et Prade [DP88] ont proposé un compromis, une combinaison mixte donnée par :

$$m(A) = \sum_{B_1 \bigcap \ldots \bigcap B_n = A} \prod_{j=1}^{n} m_j(B_j) + \sum_{B_1 \bigcup \ldots \bigcup B_n = A} \prod_{j=1}^{n} m_j(B_j) \tag{3.87}$$

Cette combinaison suppose que le conflit provient de la non fiabilité des sources et peut être modélisé par celle-ci. Cette combinaison est donc un compromis raisonnable entre la précision et la fiabilité.

3.4.5 Sélection de descripteurs

Après avoir présenté la fusion de descripteurs, nous allons nous intéresser à la deuxième façon de transformation de descripteurs, qui est la "sélection". La sélection de descripteurs a pour objectif de découvrir seulement les descripteurs les plus informatifs, qui sont individuellement discriminant, de faible redondance, pour réduire le problème de sur-apprentissage et d'augmenter la vitesse de calcul [Tor03].

Il existe trois grandes classes de méthodes de sélection : les «méthodes intégrées» (*embedded*), les «méthodes symbiose» (*wrapper*) et les «méthodes de filtre» (*filter*) [BL97, Guy03, KJ97], qui seront traitées ci-dessous :

3.4.5.1 Méthodes intégrées *"Embedded"*

Elles consistent à utiliser directement le système d'apprentissage dans l'espoir de découvrir automatiquement les descripteurs utiles pour la classification. Par exemple, un système d'induction d'arbre de décision [CM02] effectue une sélection automatique des descripteurs en choisissant ceux qui sont suffisants pour la construction de l'arbre. Malheureusement, ce type d'approche est condamné à produire des résultats peu fiables lorsque les données ne sont pas assez nombreuses par rapport au nombre d'attributs.

3.4.5.2 Méthodes symbiose *"Wrapper"*

Elles évaluent les sous-ensembles d'attributs en fonction des performances des méthodes de classification qui les utilisent. En effet, avec une méthode de classification (e.g. un perceptron multi-couches) et un ensemble d'attributs Γ, la méthode symbiose explore l'espace des sous-ensembles de Γ, utilisant la validation croisée pour comparer les performances des classifieurs entraînés sur chaque sous-ensemble, comme le montre la Fig. 3.24. Deux stratégies peuvent être utilisées : Ascendante «*forward selection*» (i.e. par ajouts successifs d'attributs), ou descendante «*backward selection*» (i.e. par retraits successifs d'attributs). Intuitivement, les méthodes symbiose présentent l'avantage de sélectionner les sous-ensembles d'attributs pertinents qui permettent les meilleures performances en généralisation, ce qui est souvent le but final. Cependant, récemment, il a été souligné que cette approche pouvait être biaisée et trop optimiste sur le vrai contenu informatif des attributs sélectionnés [AM02, XJK01]. Le principal inconvénient de ces méthodes est leur coût calculatoire attaché à l'exploration de l'espace des sous-ensembles de Γ.

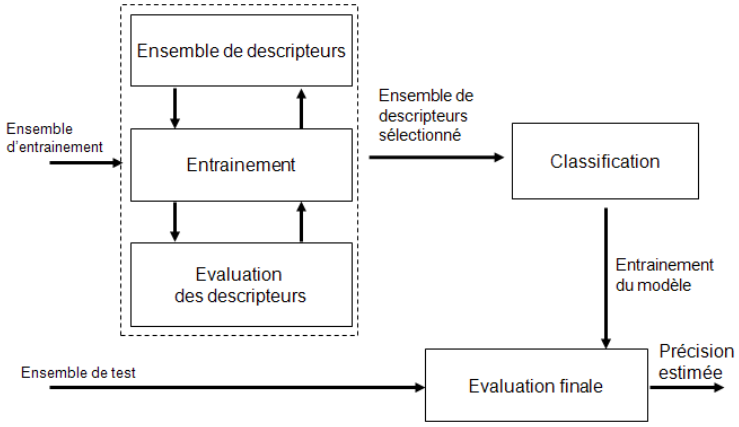


FIG. 3.24 – Représentation de la sélection des descripteurs par la méthode symbiose.

3.4.5.3 Méthodes par filtre *"Filter"*

Elles sont utilisées dans une phase de pré-traitement, indépendamment du choix de la méthode de classification. La plupart d'entre elles évaluent chaque attribut indépendamment en mesurant la corrélation (selon une métrique à définir) de leurs valeurs comme par exemple *la corrélation de Pearson, divergence de Kullback-Leibler, l'information mutuelle,* etc [LS03, Tor03]. En d'autres termes, ces méthodes évaluent l'information apportée par la connaissance de chaque attribut sur la classe des exemples. Sous certaines hypothèses d'indépendance et d'orthogonalité, les attributs ainsi estimés comme informatifs peuvent être optimaux par rapport à certains systèmes de classification. Un avantage important de cette approche est son faible coût calculatoire, puisqu'elle ne requière qu'un nombre d'évaluations linéaires, plus une opération de tri [KJ97]. Nous allons présenter brièvement certains algorithmes utilisés dans cette stratégie [Jou03].

1. **Algorithme FOCUS** introduit par Almuallim et al. [AD91, AD94] examine tous les sous-ensembles de descripteurs, puis sélectionne le plus petit sous-ensemble qui est suffisant pour déterminer l'appartenance à une classe de toutes les instances dans l'ensemble d'entraînement. Cette technique prend en compte le critère *MIN-FEATURES bias* [18]. Originellement définie pour des données booléennes non bruitées, l'algorithme est restreint à 2 classes. De plus Dash et al. [DLY97] indiquent que l'algorithme prend du temps si la taille du sous-ensemble reste importante. FOCUS présente une complexité en temps de $O(N^M)$, avec M attributs sélectionnés parmi les N du départ.
2. **Algorithme Relief** [KR92] attribue un poids à chaque descripteur pour voir son intérêt par rapport à un concept cible. Il échantillonne au hasard dans l'ensemble d'apprentissage et remet à jour la pertinence des valeurs basées sur la différence entre l'instance sélectionnée et les deux cas : proche ou non de la classe. Souvent, présenté comme une méthode résistante au bruit et de faible complexité. Cependant, il ne tient pas compte d'une éventuelle redondance entre les variables ou d'une forte corrélation. Cette méthode dépend fortement du nombre d'exemples par classe.
3. **Analyse de Variance "Anova"** [GS00] est un test statistique permettant de définir l'influence d'un ou de plusieurs attributs, en partant de l'hypothèse suivante : *l'espérance est la même pour toutes les classes* (ce qui est une hypothèse forte et non vérifiée dans nos données). D'abord, on suppose que pour chaque classe, les attributs suivent une loi gaussienne de même variance σ. Puis, on compare σ avec la variance interclasse (i.e. la variance entre les moyennes rencontrées pour chaque classe). On obtient ainsi pour chaque élément un nombre mesurant la corrélation statistique avec la classe.

Ainsi, après avoir présenté dans cette partie les différentes méthodes de transformation des descripteurs à travers la fusion et la sélection des caractéristiques bas-niveau. La prochaine partie exposera les résultats des expériences conduites sur deux bases de données.

[18] Si deux fonctions sont compatibles avec les exemples d'entraînement, préférez la fonction qui implique moins de descripteurs d'entrée.

3.5 Evaluation de la fusion bas-niveau

Dans cette partie, nous allons effectué une série d'expérimentations pour étudier l'effet d'une intégration de la fusion bas-niveau (*précoce*) sur les systèmes présentés par la Fig. 3.25 et Fig. 3.32 issues des deux projets précédemment décrits.

3.5.1 Base de données vidéos de football

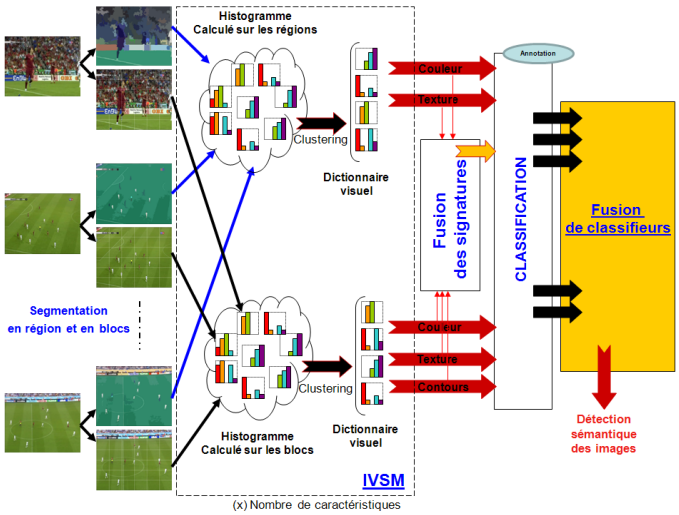


FIG. 3.25 – Schéma globale du système.

Ici, nous détaillerons rapidement le fonctionnement de notre système d'indexation dans le cadre du projet CRE-Fusion avec Orange-France Télécom. Nous discuterons des caractéristiques utilisées sur les deux méthodes de segmentation en régions et en blocs pour analyser les propriétés locales de l'image-clé présentant des scènes de football (Fig. 3.26). Deux types de descripteurs seront utilisés, la couleur (HsvHistogram, RgbHistogram) et la texture (GaborHistogram, EdgeHistogram). Un problème d'excédent d'informations se pose alors. En effet, une image contient en moyenne soixante régions et chacune d'entre elles est décrite par un vecteur de taille moyenne 100. La dimension du descripteur devient alors rapidement énorme et les opérations de comparaison complexes et fastidieuses. Nous proposons de travailler sur une approche basée sur les vecteurs de dénombrement qui sera appelée modèle vectoriel appliqué aux images (IVSM pour « Image Vector Space Model »). C'est une représentation répandue dans le domaine de l'indexation de documents textuels sous le nom de modèle vectoriel (ou « vector space model »). Il s'agit de représenter un document par le dénombrement des mots d'un dictionnaire défini au préalable. Dans le cas présent, les régions sont assimilées à des mots décrivant le contenu de l'image.

Ensuite, nous avons crée une signature par caractéristique. Quatre expériences ont été réalisées comme le montre la Table 3.6. On aura par comparaison au *Système 1* sans fusion de signatures, un système avec une fusion statique par la concaténation, et deux autres

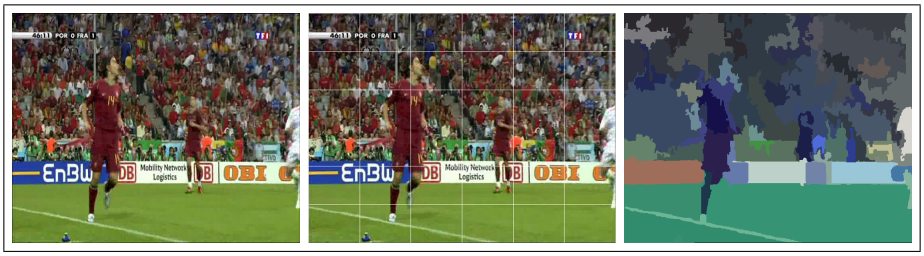


FIG. 3.26 – La segmentation en blocs et en régions d'une image-clé.

Id	Systèmes d'expériences
1	Système sans fusion de signatures (voir Fig. 3.25).
2	Système avec la fusion statique par concaténation.
3	Système avec la fusion dynamique par l'ACP.
4	Système avec la fusion dynamique par le NNC.

TAB. 3.6 – Systèmes expérimentaux.

dynamiques par l'ACP et le NNC [19].

Toutefois, il est intéressant d'examiner le comportement individuel des signatures dans la classification par SVMs. Nous rappelons que notre objectif est de construire un modèle générique capable de répondre à nos requêtes sans l'intervention d'éléments extérieurs. La Table. 3.7 donne un aperçu rapide de ce comportement. On observe que les histogrammes de couleurs RGB et HSV n'ont pas été efficaces individuellement, en utilisant la segmentation en blocs et en régions. En effet, le terrain vert et les gradins en noirs apparaissent dans la majorité des concepts, limitant ainsi le degré de discriminance de la couleur par rapport aux concepts. Par ailleurs, la texture et les contours (GabH, EDH) sont plus pertinents à l'ensemble des concepts, présentant des résultats nettement meilleurs. Deux types de descripteurs EDH ont été utilisés, le premier représente la fréquence et l'orientation des changements de luminosité dans l'image, essentiellement sur 5 types de bords dans 16 sous-images, générant ainsi un histogramme de 80 bins, alors que le deuxième ajoute un niveau global et semi-global sur la localisation des contours dans l'image, pour un total de 150 bins.

Segmentation	Bloc				Région				
Eval.(%)	RGB	HSV	GabH	EDH	RGB	HSV	GabH	EDH_{80}	EDH_{150}
MAP	6.5	7.34	**39.76**	**36.23**	7.52	15.81	**37.29**	30.93	**32.99**
F-meas	18.74	20.89	54.89	48.03	19.33	30.41	49.63	48.36	45.38
CR^+	17.90	20.73	55.81	47.81	17.66	30.89	50.05	47.32	46.71
BER	43.90	42.69	23.82	27.97	43.55	37.25	26.72	28.27	28.43

TAB. 3.7 – Résultats de la classification individuelle des descripteurs issus de la segmentation en blocs et en régions.

[19] Nous avons choisi d'utiliser le même protocole d'évaluation que celui proposé par TRECVid. La précision est calculée sur les 100 premières images retournées.

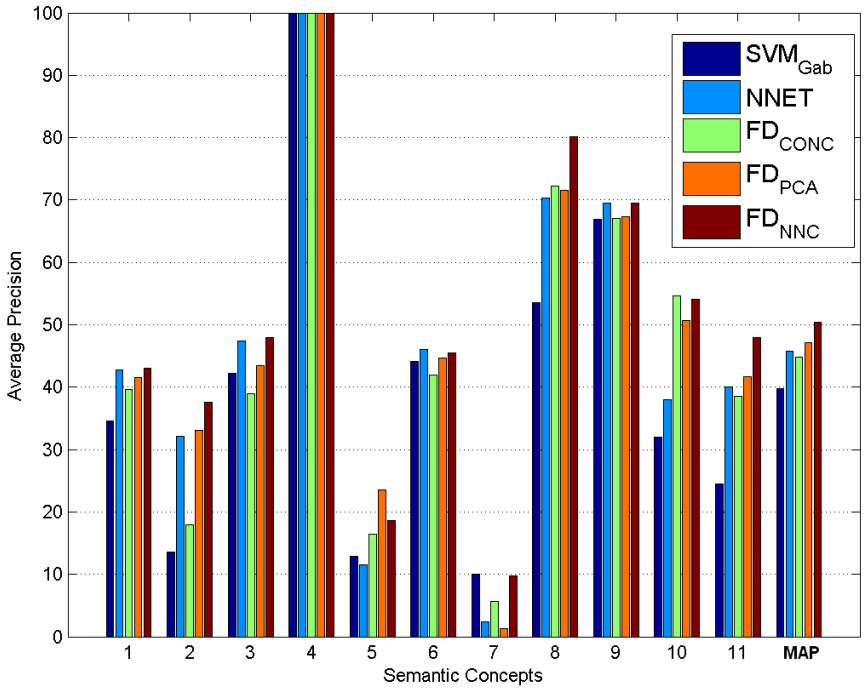


FIG. 3.27 – Comparaison de performances des quatre systèmes expérimentaux avec le meilleur résultat de la classification SVM_{GabH}.

La Fig. 3.27 présente l'évolution de la précision moyenne pour les quatre expériences. La première remarque revient à la classe (4) GOAL CAMERA qui obtient une $AP = 100\%$ (avec et sans la fusion bas-niveau), à cause d'un avantage de taille, du fait qu'il est représenté efficacement par le descripteur texture et de contours, en effet, les images de cette classe sont semblables, et contiennent des contours si caractéristiques, à savoir les hexagones du filet, comme le montre la Fig. 3.28. Dans d'autres classes, ce sont les descripteurs de couleurs qui semblent assurer un fort taux de reconnaissance. Par exemple, pour la classe (2) GAME STOP, dont la précision à progresser après la fusion (réduction de la dimension), pour atteindre 37.52%, en utilisant le NNC. Dans ce cas, toutes les images comportant l'arbitre sont bien classées. Nous pouvons aisément supposer que cela est dû aux pixels jaunes vifs du maillot de l'arbitre comme le montre la Fig. 3.29. Ainsi, les descripteurs de couleurs sont donc tout aussi indispensables que la texture et les contours. Cependant, le taux de reconnaissance est extrêmement bas pour la classe (7) GLOBAL RIGHT VIEW, pour laquelle la précision avoisine 10% en moyenne. En effet, les images de cette classe sont parfois reconnues à tort comme des images de la classe (3) LATERAL CAMERA ou de la classe (5) GLOBAL CENTER VIEW, ou même de classe (8) GLOBAL LEFT VIEW. Ceci est probablement dû à la présence d'éléments communs dans ces images, comme les gradins par exemple. Néanmoins, cela montre aussi que la présence d'éléments caractéristiques, comme les corners ou les tracés du

terrain, demeurent difficiles à exploiter. Par ailleurs, les mêmes performances sont obtenues pour les classes contenant des images très semblables avec des caractéristiques uniques (e.g. (6) Rear view et (9) Zoom on public).

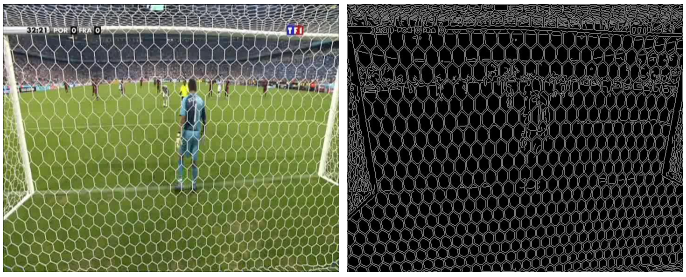


Fig. 3.28 – Exemple d'une image-clé montrant le concept Goal Camera et sa représentation des contours.

Fig. 3.29 – Images-clés représentant le concept Game Stop.

Pour le reste, la précision est représentative de la complexité des concepts, où le MAP oscille au tour de 44.31% via la concaténation (*Système 2*), ce qui est un résultat correct connaissant la complexité des concepts considérés, améliorant ainsi la classification individuelle ($MAP \cong 39\%$), et approchant les performances du *Système 1* sans fusion bas-niveau (i.e. le *Système 1* introduit les differentes sorties de classifieurs SVM obtenues par descripteurs et par concepts, dans le fusionneur NNET), qui obtient $MAP = 45.82\%$. La Fig. 3.30 présente la variation du MAP obtenu par les deux fusions dynamiques : ACP et NNC *vs* la dimension (ou $dim \in [50, 450]$ par un pas de 50). Le FD_{NNC} [20] améliore les résultats globaux de la précision à partir d'une dimension supérieure à 150, pour atteindre le maximum $MAP_{NNC} = 50.37\%$ pour une dimension de 400. Cela dit, dans cet intervalle, le gain reste faible et une dimension de 150 donne 47.34% de précision, expliquant ainsi 94% de la variance globale. Contrairement à la FD_{ACP} [21] qui présente des résultats faibles pour une dimension inférieure à 200, pour égaler un $MAP_{ACP} = 47.16\%$ à $dim = 350$. Les petites dimensions provoquent une perte de l'information, ce qui explique les faibles scores dans ces plages, alors que les grandes dimensions produisent un temps de calcul très élevé sans

[20] FD_{NNC} : Fusion bas-niveau par le NNC.
[21] FD_{ACP} : Fusion bas-niveau par l'ACP.

être sure de la qualité du nouveau vecteur (*merged vector*). Il est à noter que plusieurs possibilités sur les fonctions d'activation du réseau NNC ont été utilisées "sigmoïde, tangente hyperbolique et linéaire". Le résultat est globalement le même, avec une grande variation au niveau du temps de calcul. Nous avons préviligié la solution la plus rapide. Le choix se porte sur un NNC avec une fonction d'activation sigmoïde pour toutes les couches du NNC. Plus de détails et d'expérimentations sur ce point seront donnés dans la prochaine partie. Plusieurs concepts seront utilisés ainsi qu'une dimension de descripteurs cinq fois plus grande et une large base de données.

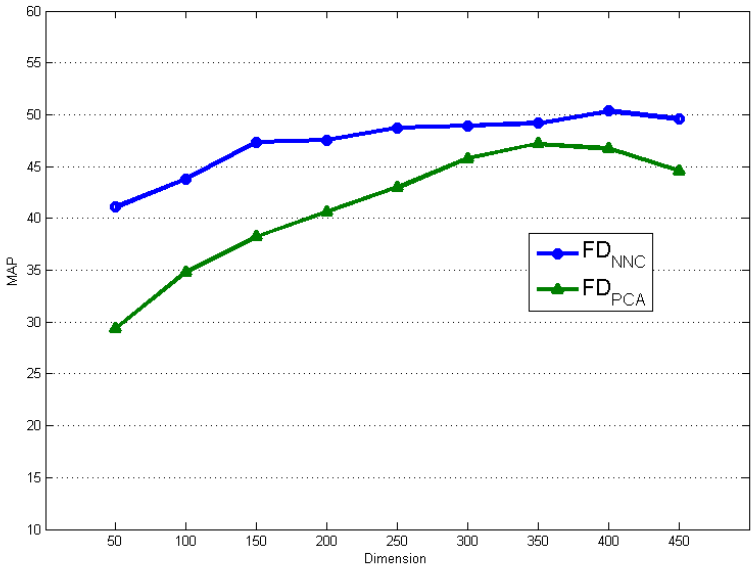


FIG. 3.30 – Evolution du MAP en fonction de la dimension pour les systèmes 3 et 4.

Par conséquent, la couche cachée du NNC à jouer le rôle d'un goulet d'étranglement dont seule l'information pertinente ressortira, combinée et transformée de manière non linéaire, pour faire coïncider les variations des données en entrée avec les variations de la quantité à prédire. Le passage par un codage permet d'avoir une nouvelle description de l'information plus compacte, riche et plus représentative.

La Table 3.8 résume l'ensemble des résultats obtenus par les quatre systèmes expérimentaux. On remarque qu'un système à base d'une fusion précoce via une réduction de la dimension est plus efficace qu'un système présentant une fusion tardive (*Système 1*) en matière de précision moyenne sur les 100 premières images-clés retournées. Le NNC à produit le score le plus élevé avec un $MAP = 50.37\%$, tout en affinant les autres statistiques globales sur l'ensemble de la base de test, en matière de F-meas, CR^+ et de BER pour un seuil de précision fixe de 0.5.

Systèmes	1	2	3	4
MAP(%)	45.82	44.31	47.16	**50.37**
F-meas(%)	56.13	53.50	55.64	61.03
CR$^+$(%)	62.12	48.59	51.98	60.30
BER(%)	20.80	26.57	25.31	21.12

TAB. 3.8 – Performances des quatre systèmes expérimentaux.

3.5.1.1 Etude statistique

Les principaux indicateurs de la qualité d'une méthode de description sont les caractéristiques des éléments du vecteur produit (e.g. les bins de l'histogramme de couleur). Les caractéristiques peuvent être mesurées par la variance inter-vecteurs, la proximité entre les éléments du descripteur, la distribution des éléments du vecteur quantifié, etc. Dans notre travail, trois méthodes ont été utilisées : (1) l'extraction d'indicateurs statistiques (la moyenne [22] et l'écart-type [23]) inter vecteurs, (2) l'analyse factorielle et (3) la classification hiérarchique ascendante (*Hierarchical Cluster Analysis*).

La moyenne et la déviation standard donnent une impression générale sur le vecteur de données (descripteur). L'indice de moyenne caractérise la situation "moyenne" de la méthode d'extraction, alors que la déviation standard donne un indice sur la qualité de discriminance. Si la déviation standard est proche du zéro, cela revient à dire que la description génère la même information pour n'importe quel type de description, ce qui la rend non-discriminante, et vice versa.

La deuxième méthode est **l'analyse factorielle** à travers l'ACP par exemple. Elle est utilisée pour éliminer la redondance dans les vecteurs de données par l'identification des facteurs qui représentent sans grande perte la variance globale de l'information.

Enfin, une autre méthode pour détecter la redondance est proposée par la méthode **classification hiérarchique ascendante** (CHA), où pour un niveau de précision donné, deux éléments peuvent être confondus dans un même groupe, alors qu'à un niveau de précision plus élevé, ils seront distingués et appartiendront à deux sous-groupes différents produisant un arbre appelé "dendogramme". Initialement, la CHA considère toutes les observations comme étant des clusters ne contenant qu'une seule observation (singleton), et leur distance est alors le plus souvent définie comme étant Euclidienne. La première étape consiste donc à réunir dans un cluster à deux observations, les deux plus proches. Puis la CHA continue, fusionnant à chaque étape les deux clusters les plus proches au sens de la distance choisie. Le processus s'arrête quand les deux clusters restant fusionnent dans l'unique cluster contenant toutes les observations.

1. **Avant la fusion bas-niveau :** Le calcul de la moyenne donne une valeur de "0.45", l'écart type est inférieur à "0.1" (les valeurs sont normalisées entre $[0, 1]$). Cela re-

[22] La moyenne est une mesure simple, utile, mais aussi trompeuse. Elle n'est réellement significative que sur un grand nombre de valeurs. Pour en savoir plus, il faut connaître la répartition des valeurs autour de la moyenne, comment elles se distribuent.

[23] l'écart type (*standard deviation* en anglais) mesure la dispersion d'une série de valeurs autour de leur moyenne.

vient à dire que la description génère une même information qui peut être qualifiée de non-discriminante. Ce qui signifie que les valeurs des signatures pourraient être transformées et quantifiées dans un type de données de façon à augmenter le degré de discriminance.

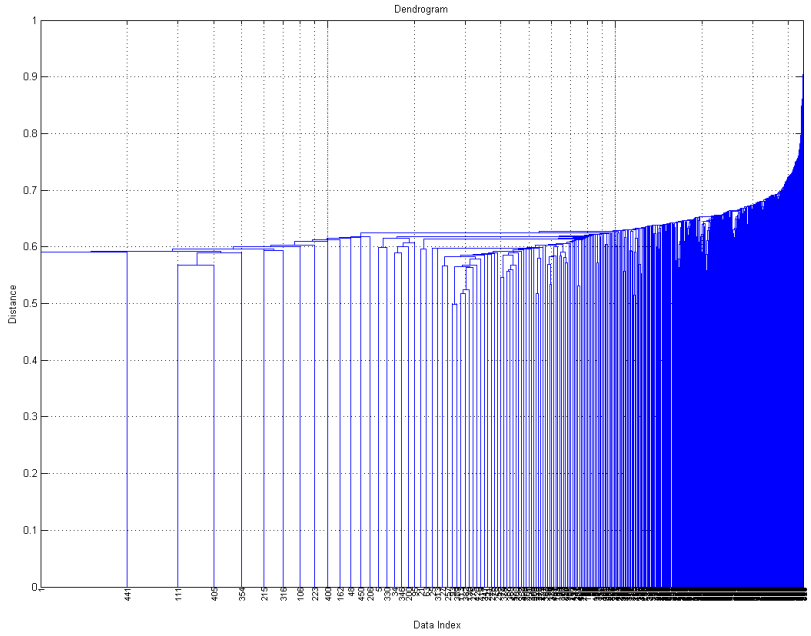


FIG. 3.31 – Regroupement par la CHA.

La CHA présente de manière intuitive la proximité entre les éléments des signatures, donnant une idée sur l'homogénéité interne la plus élevée. Au début, elle regroupe les éléments à forte similitude de couleur, en particulier ceux du RGB puis HSV dans les deux types de segmentation. La distance est inférieure à 0.6 comme le montre les liens hiérarchiques qu'apparaissent sur le dendogramme de la Fig. 3.31, puis, les éléments avec de fort histogrammes de contours EDH_{80} et EDH_{150} (il est à noter que les 80 premiers bins des deux EDH sont identiques et représentent la fréquence et l'orientation des changements de luminosité dans l'image, essentiellement sur 5 types de bords dans 16 sous-images) puis ceux de GabH. Enfin, il est facile de voir que les descripteurs de couleurs et de contours sont les plus redondants.
Par ailleurs, un phénomène intéressant est observé pour le EDH. Indépendamment de la nature de la segmentation ou le type du média, Eidenberger [Eid03] constate que EDH ne donne pas de mesures dans les intervalles $[0.32, 0.42[$, $[0.72, 0.81[$, $[0.9, 1[$. Ceci est expliqué par le simple comptage des contours dans certaines orientations prédéfinies dans une région. Ainsi, 30% de l'intervalle alloué aux données de ce descripteur ne sont pas utilisés. Pour cela, il est souhaitable de transformer et de quantifier ce dernier pour en réduire le problème de sauvegarde.

2. **Après la fusion bas-niveau :** Les nouveaux écarts-types sont de "0.204" pour la réponse du ACP et de "0.373" pour le NNC en utilisant une dimension $dim = 350$. Dans la Fig. 3.30, nous pouvons remarquer qu'il suffit d'utiliser 150 facteurs pour être capable d'expliquer 94% de la variation globale. Les cinquante premiers facteurs a eux seuls (i.e. $dim = 50$) expliquent 58% de cette variance utilisant l'ACP, et 82% avec le NNC, ce qui montre que les descripteurs avant fusion sont redondants. Ces résultats ne sont pas surprenants car quatre des neuf descripteurs (avec les deux types de segmentation) utilisés dans nos expériences représentent la couleur et le reste la texture.

3.5.2 Base de données TRECVid 2007

Les expérimentations ont été reconduites dans TRECVid 2007 vidéos [TRE], qui est un laboratoire d'évaluation approchant les situations de difficulté qu'on trouve dans le monde réel. Nous nous plaçons dans les mêmes conditions que les expériences décrites dans la section. 3.3.2, en utilisant cette fois-ci, seulement cinq descripteurs MPEG-7 globaux d'une dimension totale concaténée de 736, sur les 11 descripteurs du départ : ColorLayout (CLD, 12 bins), ColorStructure (CSD, 256 bins), ScalableColor (SCD, 256 bins), EdgeHistogram (EHD, 150 bins) et HomogeneousTexture (HTD, 62 bins). Ce choix est motivé par un souci de capacité calculatoire, afin de permettre une évaluation sur l'ensemble des 36 concepts (voir la Fig. 3.32).

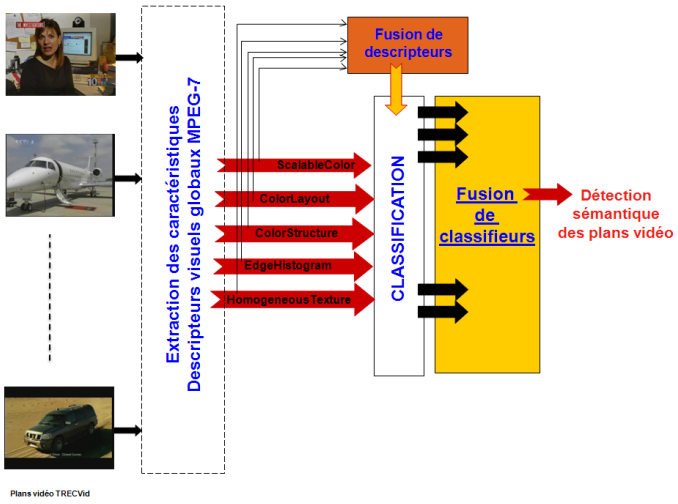


FIG. 3.32 – Schéma globale du système.

Comme décrit dans la section 2.1.1, un bon descripteur est celui qui présente une grande variance et produit des valeurs uniformément distribuées. Les indicateurs statistiques peuvent donner une idée sur la qualité de nos descripteurs. Par conséquent, on pourra prétendre que le descripteur ColorLayout va être négatif pour notre système, car aucun des 12 bins n'a de déviation standard σ qui dépasse 0.036, avec une moyenne μ de 0.48. Le

même constat a été obtenu pour ColorStructure avec $\sigma = 0.043$ et $\mu = 0.23$, contrairement à ScalableColor qui présente $\sigma = 0.28$ pour 24 bins avec $\mu = 0.42$.

EdgeHistogram présente de façon excellente les données avec plusieurs bins de σ supérieure à 0.3 et une μ proche de 0.5. Pour HomogenousTexture, les mesures sont relativement acceptables pour les plans monochromes ($\sigma = 0.18$, $\mu = 0.23$).

Par ailleurs, les descripteurs de couleurs sont globalement bons pour les plans couleurs. Le faible σ est due au nombre important de plans monochromes, qui fait baisser la description générale du descripteur couleur. Contrairement à la texture qui est très sensible aux plans monochromes que couleurs, sauf pour EdgeHistogram. Ce dernier se place dans notre analyse comme le meilleur descripteur (ce qui confirme et vérifie nos précédents résultats de la section. 3.3.1, où la classification avec EDH a donné les meilleures résultats).

Concernant la question de savoir quels sont les descripteurs qui peuvent être compactés ou réduits ? Dans un premier temps, nous allons reprendre les mêmes remarques précédemment citées par Eidenberger [Eid03], sur la non existante de données dans certains intervalles de descripteurs. En effet, EdgeHistogram n'a pas de mesure dans les intervalles $[0.32, 0.42[$, $[0.72, 0.81[$, $[0.9, 1[$ et pour ScalableColor dans les intervalles $[0.1, 0.2[$, $[0.4, 0.5[$, $[0.7, 1[$ à l'exception sur les 16 premiers bins. Ceci est expliqué par l'application de la transformation de Haar, où seulement 45% de la capacité de stockage peut être utilisée en compactant ce descripteur. Enfin, la plupart de l'énergie et de la déviation standard observées dans HomogeneousTexture est dans $[0.5, 1[$. Un gain de 50% en stockage peut être obtenu.

La Fig. 3.33 montre la variation du MAP *vs* la dimension, par pas de 100, pour 5 variantes [24] du NNC (selon les fonctions d'activations utilisées) et l'ACP.

Intéressons nous d'abord aux 5 résultats du NNC. On remarque que plus le nombre de dimensions augmente, plus le système intègre de l'information et le MAP progresse. Pour $dim = 500$, on obtient des résultats semblables pour l'ensemble des possibilités NNC, qui convergent vers $MAP = 14\%$, expliquant ainsi 95% de la variation globale. Or, une dimension inférieure à 500 conduit à une grande perte de précision. Le résultat maximal du $MAP = 14.66\%$ a été obtenu pour $dim = 600$ avec le NNC_{st}. La réduction de la dimension a permis une amélioration visible des performances. Concernant l'ACP, le résultat ne dépasse pas $12,5\%$, avec une stabilisation des performances pour $dim \in [500, 700]$, mais il présente un temps de calcul plus faible parsqu'il n'effectue pas d'apprentissage. Par ailleurs, nous pouvons remarquer qu'il suffit d'utiliser 100 facteurs pour être capable d'expliquer 80% de la variation globale de l'ACP et du NNC, avec des deviations standards moyennes de 0.1329 et 0.1474 respectivement. Ceci montre que les descripteurs avant fusion bas-niveau sont très redondants.

[24] Les propriétés de la fonction d'activation influent sur celle du neurone et il est donc important de bien choisir celle-ci pour obtenir un modèle utile en pratique. Dans notre étude, un programme se charge d'essayer toutes les architectures en faisant varier le nombre de neurones cachés et les fonctions d'activations utilisées, pour en retenir que la plus performante :

- **sl** : on utilisera une fonction sigmoïde **s** dans la couche cachée, et linéaire **l** pour la couche de sortie.
- **st** : sigmoïde **s** dans la couche cachée, tangente hyperbolique **t** pour la couche de sortie.
- **ss** : sigmoïde **s** pour les deux couches.
- **tt** : tangente hyperbolique **t** pour les deux couches.
- **ts** : tangente hyperbolique **t** dans la couche cachée, sigmoïde **s** pour la couche de sortie.

Enfin, nous pouvons relever trois points : (1) la réduction de la dimension a été positive pour notre système, (2) le NNC a obtenu de meilleures performances par rapport à l'ACP, (3) en comparaison avec nos précédents résultats obtenus dans la fusion de classifeurs (haut-niveau). Nous pouvons dire qu'un système intégrant une fusion dynamique de descripteurs (bas-niveau) à base du NNC peut être plus efficace qu'un système avec la fusion de classifeurs (haut-niveau). Cette fusion précoce a réussi à apprendre, à sélectionner de l'hétérogénéité des caractéristiques, les informations nécessaires pour mieux décider et agir dans l'étape de classification.

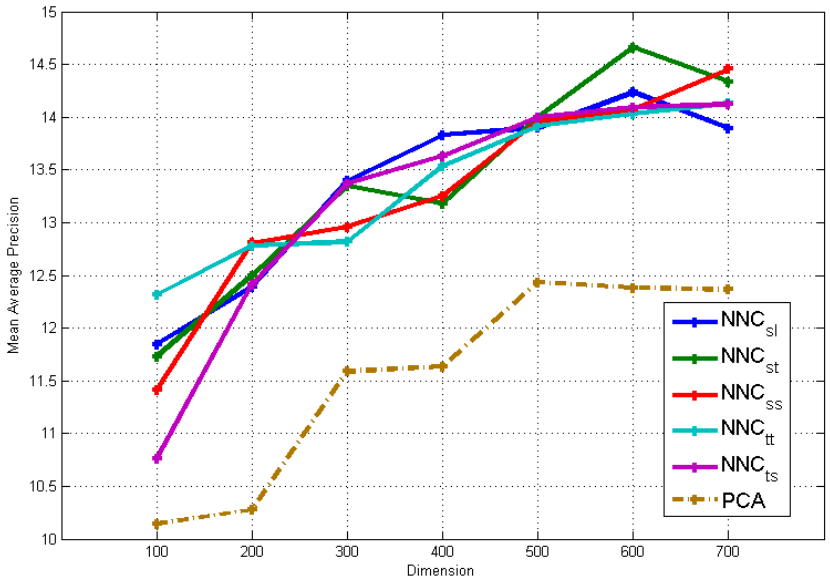


FIG. 3.33 – Evolution du MAP en fonction de la dimension, pour le schéma à base d'une ACP et des différentes possibilités du NNC.

Intéressons nous maintenant aux concepts qui ont pu être améliorés pour permettre cette performance. La Fig. 3.34 présente les résultats de la classification après une étape de fusion de descripteurs par NNC_{st} en utilisant une $dim = 600$, avec ceux du PENN. Nous remarquons deux choses : (1) les concepts rehaussés ne sont pas uniquement ceux qui présentent de fortes caractéristiques visuelles, soit en couleur ou en texture tels que SPORTS, OUTDOOR, BUILDING, VEGETATION, ROAD, SNOW, SKY, URBAN, WATERSCAPE, BOAT/SHIP, MAPS mais on retrouve aussi des concepts plus abstrait comme MEETING et STUDIO. (2) Une perte de précision est visible sur CROWD, FACE, PERSON, WALKING/RUNNING, PEOPLE MARCHING, ceci est tout simplement due à la non utilisation des descripteurs spécifiques tels que FaceDetector, CameraMotion, MotionActivity, etc, qui permettaient une détection plus précise de ces classes par le PENN.

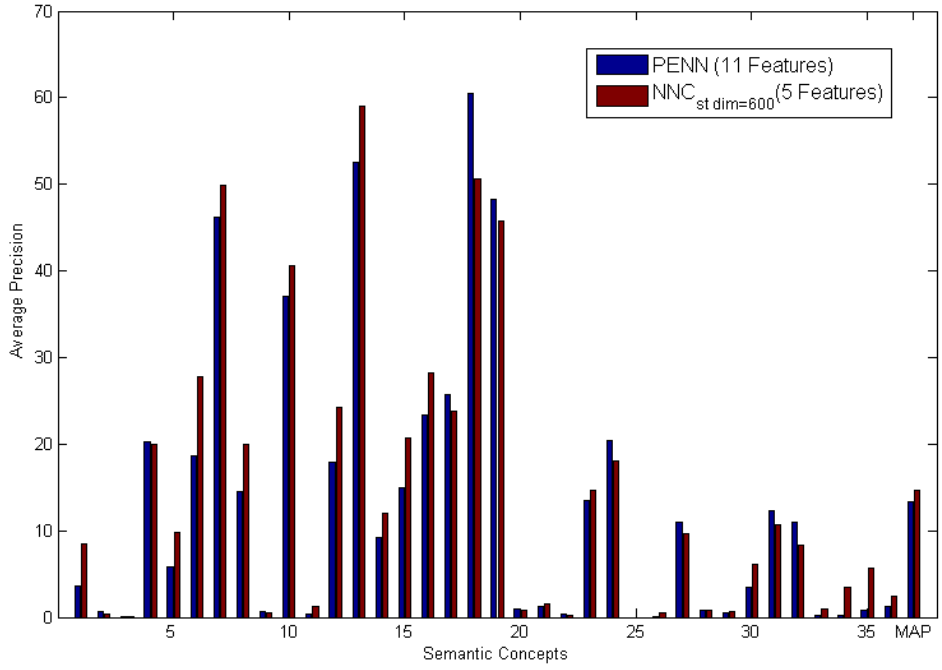


FIG. 3.34 – Comparaison entre les résultats obtenus par le PENN (avec 11 descripteurs) et NNC_{st} de dimension 600 (avec 5 descripteurs), par concepts.

3.6 Conclusion

L'état de l'art montre que la combinaison parallèle des classifieurs (fusion haut-niveau) est une voie de recherche prometteuse, qui permet l'amélioration des systèmes de reconnaissance. Les travaux effectués sur la combinaison dévoilent aussi la multitude de techniques qui diffèrent par leurs capacités d'apprentissages et le type de sortie des classifieurs.

Le réseau de neurones reste celui qui donne le meilleur résultat comparant aux méthodes avec et sans apprentissage développés dans notre travail. Le WBF a amélioré pour certains concepts la précision, en résolvant le problème du sur-apprentissage causé par le manque de données. Par ailleurs, l'introduction de la théorie de évidences et particulièrement le NNET a permis de résoudre certaines ambiguïtés, à travers les informations apportées par l'incertitude et l'ignorance du système. Dans le même raisonnement, le PENN a permis de modéliser deux étapes dans la même tache de combinaison, qui sont : la pondération des sorties de classifieurs à travers l'étude de la perplexité, qui estime le degré de relation entre les descripteurs et les concepts, avec la combinaison NNET. La fusion de classifieurs relève une nouvelle fois son importance.

Par ailleurs, la fusion joue un rôle primordial dans la classification, et semble être plus efficace en amont des systèmes de classifications qu'en aval. Nous constatons que 81% des

composantes sont conservées par le système le plus pertinent qui est le NNC_{st}. Les propriétés de la méthode sont mises à profit pour avoir une meilleure réduction de dimensionnalité, tout en soulignant la forte corrélation entre la quantité des données et les performances du système. Cependant, cette tache est longue et fastidieuse mais ouvre une porte et des perspectives pour ce champs de recherche.

Nous allons apporter une attention toute particulière aux systèmes génériques dont l'entraînement doit s'adapter à une grande variété de contenus et de concepts. Dans le prochain chapitre, nous présenterons comment modéliser les relations entre les concepts sémantiques, et de quelle manière cette information peut être introduite dans notre système d'indexation et de recherche de plans vidéo, pour une prise de décision efficace.

Chapitre 4

L'ontologie et la Similarité Inter-concepts

La plupart des modèles d'indexations sont basés sur la classification binaire, en ignorant toutes les relations possibles entre les concepts. Cependant, les concepts n'existent pas en isolation et sont reliés par leurs interprétations sémantiques et la cooccurrence. Deux difficultés doivent être prises en compte. La première réside dans l'utilisation d'une ontologie qui décrit les relations existantes entre les concepts. La deuxième est liée à l'exploitation de cette information sémantique par les systèmes de classification ou de fusion.

Dans ce chapitre, nous proposons un système ontologique d'indexation particulièrement pour l'étape de raisonnement et de la construction de la décision par un réajustement des valeurs de confidences issues du PENN (Fig. 4.4). On se focalisera sur l'extraction et l'exploitation du contenu à haute valeur sémantique et par conséquent de l'interprétation globale du plan vidéo. Pour cela, une étude de la similarité inter-concepts est effectuée en introduisant 3 types d'informations : les descripteurs bas-niveau, la cooccurrence et la similarité sémantique issue de l'approche hybride. Par ailleurs, pour cette dernière, trois approches hybrides seront étudiées et comparées. Le système final s'appellera ***Ontological PENN "Onto-PENN"****.*

4.1 L'ontologie

En philosophie, l'ontologie est l'étude de l'être en tant qu'être (i.e. l'étude des propriétés générales de ce qui existe). Le terme est actuellement repris en sciences de l'information, pour constituer un modèle de données représentatif d'un ensemble de termes et de concepts dans un domaine donné, réel ou imaginaire. Ce modèle est organisé dans un graphe dont les relations peuvent être des relations sémantiques et des inclusions.

On retrouve les ontologies dans plusieurs domaines tels l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale et l'architecture de l'information comme une forme de représentation de la connaissance au sujet d'un monde ou d'une partie de ce monde.

Une des définitions de l'ontologie qui fait autorité est celle de Gruber [Gru93] :

Une ontologie est la spécification d'une conceptualisation d'un domaine de connaissance.

Parallèlement à cette définition assez théorique de l'ontologie, une autre définition, plus opérationnelle, peut être formulée ainsi :

Une ontologie est un réseau sémantique qui regroupe un ensemble de concepts décrivant un domaine, liés par des relations sémantiques et hiérarchiques.

L'ontologie peut être construite selon trois méthodes distinctes :

1. *Manuellement :* les experts réalisent une ontologie d'un domaine ou de son extension (niveaux supérieurs de Cyc [1], WordNet [2]).
2. *Automatiquement :* par des techniques d'extraction de connaissances des concepts et de leurs relations.
3. *Méthodes mixtes :* par des techniques automatiques qui permettent d'étendre des ontologies construites manuellement (base des connaissances Cyc).

L'ontologie a été historiquement utilisée pour améliorer les performances dans les systèmes de recherche multimédia [WTS04, NKFH98, FGL07]. Nous nous intéresserons dans ce qui va suivre à ces travaux, en particulier dans l'étude des relations entre concepts à travers la similarité inter-concepts. Avant cela, nous allons présenter l'ontologie LSCOM-lite utilisée dans la suite de nos travaux.

4.1.1 L'ontologie LSCOM-lite

L'ontologie LSCOM-lite est une version allégée de l'ontologie LSCOM [NKH+06]. Elle a été développée dans l'atelier ARDA/NRRC [3] qui contient les 39 concepts suivants (Sports, Entertainment, Weather, Court , Office, Meeting, Studio, Outdoor, Building, Desert, Vegetation, Mountain, Road, Sky, Snow, Urban, Waterscape, Crowd, Face, Person, Government-Leader, Corporate-Leader, Police Security, Military, Prisoner, Animal, Computer Tv, US Flag, Airplane, Car, Bus, Truck, Boat/Ship, Walking/Running, People Marching, Explosion/Fire, Natural Disaster, Maps, Charts) sur les 834 initiaux, puis des 449 concepts choisis pour la première version du LSCOM.

LSCOM-lite a été utilisé par le NIST (*National Institute of Standards and Technology*) en collaboration avec les participants TRECVid. Quelque 80 heures d'un total de 61901

[1] Le projet Cyc (qui dérive du mot «encyclopédie») lancé en 1984, cherche à développer une ontologie globale et une base de données de la connaissance générale, dans le but de permettre à des applications d'intelligence artificielle de raisonner d'une manière similaire à l'être humain [MWK+05].

[2] WordNet est une ontologie lexicographique pour la langue anglaise [Fel98]. Elle est représentée sous la forme de listes liées entre elles pour créer un réseau. Elle est utilisée pour un dictionnaire (WordWeb2), un système expert (SearchAide), un logiciel d'annotation automatique des textes, etc. WordNet 1.7 a un réseau de 144 684 mots, organisés en 109 377 concepts appelés "synsets".

[3] NRRC : Northeast Regional Research Center. http ://nrrc.mitre.org/NRRC/workshops.htm

plans vidéos issues de TRECVid'05 ont été annotées. Ces concepts sont regroupés en 6 catégories :"*location, objects, people, events, program, graphics*". Deux avantages sont à retenir de ce partage des annotations : (1) l'obtention d'une grande quantité de vidéos annotées, (2) la possibilité de comparer objectivement les différents systèmes proposés puisqu'ils sont entraînés et testés sur les mêmes vidéos. Il est à noter que les 3 concepts ENTERTAINMENT, GOVERNMENT-LEADER et CORPORATE-LEADER ont été supprimés dans la compagne d'évaluation TRECVid 2007 pour en garder que 36 concepts.

4.2 La similarité inter-concepts

Les méthodes présentées dans la section. 3.2 ont pour but la fusion des sorties de classifieurs binaires. Ces dernières sont entraînées indépendamment sur les différentes caractéristiques, sans prendre en considération les éventuelles relations entre les concepts. Or, ces derniers n'existent pas en isolation et sont reliés par leurs interprétations sémantiques et de leurs cooccurrences. Par exemple, le concept CAR cooccure souvent avec le concept ROAD, contrairement à MEETING et ROAD. La mise en relation des concepts peut inférer de nouveaux concepts plus complexes ou d'améliorer la reconnaissance des concepts préalablement détectés. Ainsi, la présence ou l'absence de certains concepts suggère une forte ou une faible possibilité de détecter d'autres concepts (e.g. la détection de SKY et de SEA augmente la probabilité du concept BEACH et réduit la probabilité de DESERT).

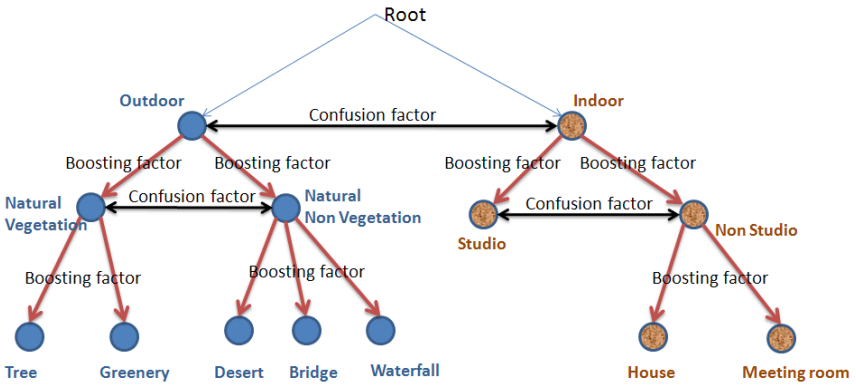


FIG. 4.1 – Exemple d'ontologie utilisée par Wu et al. [WTS04].

Dans la littérature, plusieurs approches ont été proposées, on citera les travaux de Wu et al. [WTS04] qui proposent un apprentissage d'une classification multi-classes basée sur l'ontologie [4] pour la détection de 133 concepts dans les vidéos TRECVid 2003. Cette approche s'effectue en deux étapes : chaque concept est modélisé indépendamment. Ensuite, un apprentissage à base de l'ontologie améliore la précision du modèle individuel en considérant les influences des relations possibles entre concepts. Deux types d'influences ont été définis : facteur d'amplification *boosting factor* "λ" et facteur de confusion *confusion factor* "β". Les

[4] L'ontologie est obtenue par l'outil d'annotation d'IBM VideoAnnEX [LTS03] sur 62 heures de vidéos.

facteurs sont obtenus à partir de l'étude de la corrélation des données. Puis, une mise à jour est effectuée de la nouvelle confidence $\underline{p}(x/C_i)$ comme le montre les équations 4.1 et 4.2 respectivement. Le facteur d'amplification représente l'influence entre l'ensemble des concepts ancêtres ψ et leurs descendants C_i (e.g. entre le concept ancêtre OUTDOOR et descendants DESERT, TREE, WATERFALL), alors que le facteur de confusion représente l'influence avec les concepts qui ne peuvent coexister θ (e.g. entre OUTDOOR et INDOOR). Les auteurs ont remarqué que les concepts parents présentent de meilleurs taux de classification que leurs descendants. C'est pour cette raison, l'idée du facteur d'amplification est de booster la précision des concepts en s'appuyant sur les informations issues des parents. De l'autre coté, le facteur de confusion s'appui sur le comportement des concepts de θ. Si une image (par exemple) présente une forte probabilité pour un concept donné, qui ne peut co-exister avec un autre. Alors, ce dernier sera affaibli et vice versa.

$$\begin{cases} \underline{p}(x/C_i) = p(x/C_i) + \sum_{j\in\psi} \lambda_j^i p(x/C_j) \\ \lambda_j^i = \frac{A}{B+\exp\left(C|p(s/C_i)-p(s/C_j)|\right)} \end{cases} \tag{4.1}$$

$$\begin{cases} \underline{p}(x/C_i) = \frac{p(x/C_i)}{\beta} \\ \beta = \frac{1}{f(p(x/C_i)-\max_{j\in\theta}(p(x/C_j)))} \end{cases} \tag{4.2}$$

Les paramètres A, B et C de l'équation 4.1 sont obtenus empiriquement comme dans les travaux de [LG03] sur le calcul des confidences pour une classification multi-classes. $f(.)$ est une fonction positive croissante pour l'équation 4.2.

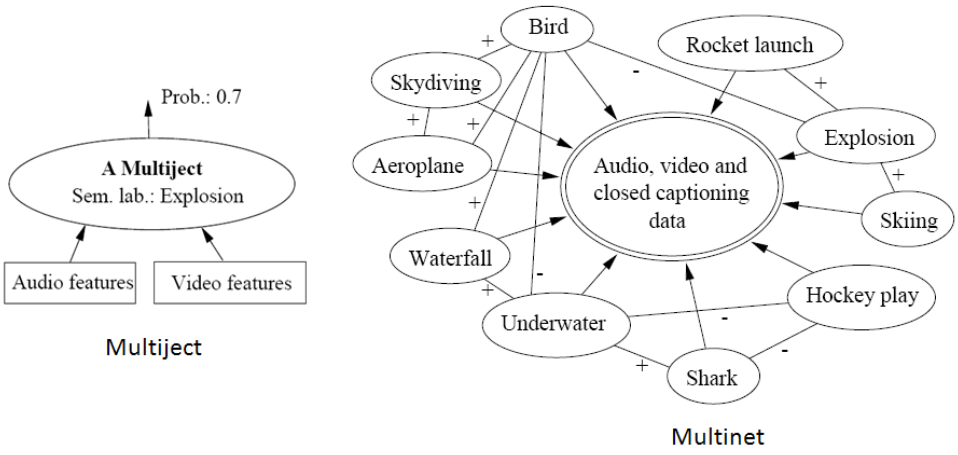


FIG. 4.2 – Modèle Multinet d'indexation du contenu [NKFH98]. Les signes positifs dans le graphe montrent les interactions positives (forte corrélation), et vice versa.

Naphade et al. [NKFH98] placent tous les concepts au même niveau sémantique, reliés par une relation de dépendance conditionnelle avec les descripteurs bas-niveau qui leurs sont associés. La Fig. 4.2 présente le modèle proposé par les auteurs à travers un réseau

de concepts [5] appelé **Multinet** (Multimedia network). L'intégration des *multijects* dans le réseau formant un graphe de probabilité associant une valeur de poids pour chaque relation (arc) dans le graphe, par un modèle statistique : réseau Bayésien [6] et les modèles de Markov. Ainsi, la phase d'apprentissage apprend les cooccurrences entre chaque *multiject*, puis un processus itératif propage ces informations dans le réseau pour mettre à jour les valeurs prédites, essentiellement par des sommations ou des multiplications des informations entrantes sur chaque *multiject*.

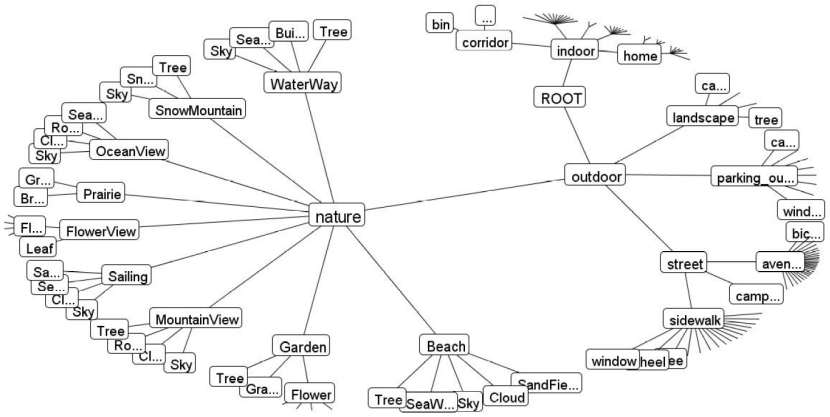


FIG. 4.3 – Structure d'ontologie représentant l'organisation hiérarchique des concepts [FGL07].

Fan et al. [FGL07] de l'université de Caroline du Nord proposent une classification hiérarchique pour l'annotation des images [7]. Cette approche incorpore les dépendances contextuelles de l'ontologie WordNet et les relations de cooccurrences, comme le montre les équations suivantes :

$$\begin{cases} \lambda(C_m, C_n) = \rho(C_m, C_n)\pi(C_m, C_n) \\ \text{ou} \quad \rho(C_m, C_n) = log\left(\frac{P(C_m,C_n)}{P(C_m)P(C_m)}\right) \\ \text{et} \quad \pi(C_m, C_n) = -log\left(\frac{dist(C_m,C_n)}{2D}\right) \end{cases} \tag{4.3}$$

avec $\rho(C_m, C_n)$ est la probabilité jointe des deux concepts, obtenue par le calcul de la fréquence de cooccurrence des termes C_m et C_n, et $\pi(C_m, C_n)$ est la dépendance contextuelle tirée de la structure ontologique (*dist* représente le chemin le plus court entre deux concepts et D est la profondeur maximale dans WordNet).

Hauptmann et al. [HCC+05] de l'université Carnegie Mellon proposent une comparaison entre une indexation multimodale et unimodale. Le système multimodal apprend en plus les

[5] Dans [NKFH98], les concepts sont appelés *Multijets* (Multimedia objects). Chaque multiject est considéré comme une variable aléatoire binaire qui est explicitement mise en relations avec les autres multijects.

[6] Un réseau Bayésien est un graphe dirigé sans circuit, dont les noeuds représentent les variables d'intérêt du domaine et les arcs représentent les dépendances entre les variables. A chaque variable est associée une table de probabilité conditionnelle.

[7] Trois bases d'images ont été utilisées : *Corel Images, Google Images et LabelMe*.

dépendances entre les différents concepts, utilisant les modèles graphiques suivants : *Conditional Random Field "CRF"* et *Réseaux Bayesien*. Les deux modèles obtiennent des résultats en termes de précision très proches mais restent supérieurs aux résultats du système unimodal. Koskela et al. [KS06] de l'université de Dublin, proposent un système de regroupement (*Clustering*) basé sur les relations entre les concepts, enrichi par des descriptions bas-niveau. Ces travaux cherchent à exploiter les corrélations entre les concepts pour réduire le fossé sémantique, à travers divers métriques pour le calcul de la similarité inter-concepts.

Dans un autre registre, Li et al. [LBM03] de l'université de Manchester construisent cette relation en se basant uniquement sur la similarité sémantique. Ils étudient plusieurs mesures de similarités non-linéaires $S = f(f_1, f_2, f_3)$ en fonction du chemin l, de la profondeur h (i.e. nombre de niveau par rapport au concept subsument CS) dans une ontologie et de la densité locale d, comme le montre l'équation 4.4. Ces points sont discutés en détails dans la section 4.4.3.

$$\begin{cases} f_1 = \exp(-\alpha l) \\ f_2 = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \\ f_3 = \frac{e^{\delta d} - e^{-\delta d}}{e^{\delta d} + e^{-\delta d}} \end{cases} \tag{4.4}$$

avec, la constante α, le paramètre de lissage β, et $\delta = max_{c \in CS(c_m, c_n)}(-\log p(c))$ qui représente la similarité sémantique mesurée par le contenu informatif. Plusieurs combinaisons ont été appliquées comme par exemple : $S_1 = f_1$, $S_2 = f_1 f_2$, $S_3 = S_2 f_3$, $S_4 = S_2 + f_3$, etc. Les résultats des comparaisons en variant les paramètres (α et β) indiquent que plusieurs fonctions présentent des résultats satisfaisants, en particulier celles qui englobent les trois influences.

Discussion

Les travaux de Wu et al. [WTS04] utilisent une mise à jour des confidences en s'appuyant sur la corrélation des données et de la structure fixe d'une ontologie. Naphade et al. [NKFH98] se basent sur l'entraînement des descripteurs bas-niveau et de la cooccurrence entre concepts. Koskela et al. [KS06] intègrent la cooccurrence et l'information visuelle dans la construction de cette relation. Fan et al. [FGL07] comme Li et al. [LBM03] incorporent les dépendances contextuelles de l'ontologie WordNet et les relations de cooccurrences. Dans cette thèse, nous nous plaçons dans la continuité pour la construction de la similarité inter-concepts. Nous utiliserons la cooccurrence dans le corpus, l'information visuelle issue des descripteurs MPEG-7 bas-niveau et enfin la structure sémantique obtenue à partir de l'ontologie LSCOM-lite. Dans ce qui va suivre, nous allons d'abord présenter l'architecture du système proposé. Ensuite, nous aborderons plus en détails les formes de similarités utilisées dans notre architecture.

4.3 Architecture du système

Cette section décrit l'architecture générale de notre système qui peut être résumer en cinq étapes : (1) extraction des descripteurs visuels, (2) classification, (3) pondération des descripteurs par concept à base de la perplexité, (4) fusion de classifieurs (NNET) et enfin

(5) réajustement des valeurs de confidence (ce que nous appelons dans la Fig. 4.4 par l'étape du raisonnement).

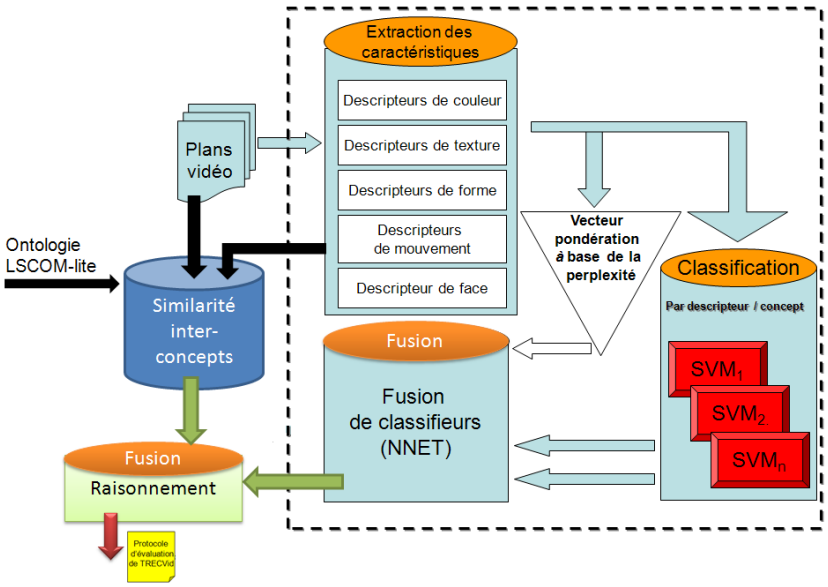


FIG. 4.4 – Architecture générale du système d'indexation.

Les quatre premières étapes ont été déjà discutées dans les chapitres précédents. Nous allons nous focaliser sur la 5ème étape (dite de raisonnement) qui est un réajustement de la valeur des confidences obtenues par le PENN. Pour cela, nous intégrons des informations sur les corrélations entre les concepts issues des descripteurs visuels bas-niveau, de la cooccurrence et de l'ontologie utilisée.

4.4 Construction de la similarité inter-concepts

Il est important de prendre en considération quelque contraintes dans la construction de la similarité inter-concepts. La psychologie démontre que la similarité dépend du contexte et peut être asymétrique [Lin98, LBM03]. Dans l'ontologie LSCOM-lite [NKK+05], on peut facilement identifier deux types de relations : (1) relations positives comme (BUILDING, OUTDOOR), (VEGETATION, MOUNTAIN), (ROAD, CAR) et des (2) relations négatives (BUILDING, SPORTS), (SKY, MEETING), (ROAD, OFFICE).

Dans cette section, nous allons décrire les formes de modélisation et d'utilisation des similarités inter-concepts. Une méthode directe pour le calcul de la similarité est de chercher le chemin minimum connectant deux concepts [RMBB89]. Par exemple, la Fig. 4.5 illustre un fragment de l'ontologie hiérarchique LSCOM-lite. Le chemin minimum entre le concept VEGETATION et ANIMAL est le suivant : "VEGETATION-OUTDOOR-LOCATION-ROOT-OBJECTS-ANIMAL" d'ou d_{path}(VEGETATION, ANIMAL) = 5. Or, le chemin minimum entre VEGETATION et OUTDOOR est d_{path}(VEGETATION, OUTDOOR) = 1. Ceci permet de dire que

OUTDOOR est plus proche sémantiquement de VEGETATION que du concept ANIMAL. Par contre, on ne peut pas dire que ANIMAL est proche de CAR, même si $d_{path}(\text{ANIMAL}, \text{CAR}) = 1$. Ainsi, cette méthode est dépendante de la structure de l'ontologie. De plus, il est clair que OUTDOOR est un concept global qui regroupe plusieurs concepts tels que "DESERT, URBAN, ROAD, etc", chacun avec une description de scène en couleurs et en textures différentes. Afin d'adresser ces difficultés, nous allons introduire plus d'informations qui représentent la relation entre les concepts comme la cooccurrence, les descripteurs visuel bas-niveau, le chemin et la profondeur dans un graphe ontologique, etc, afin de booster les performances de notre système d'indexation comme le montre l'équation suivante :

$$\lambda(C_m, C_n) = Sim_{cos}(C_m, C_n) + Sim_{vis}(C_m, C_n) + Sim_{sem}(C_m, C_n) \tag{4.5}$$

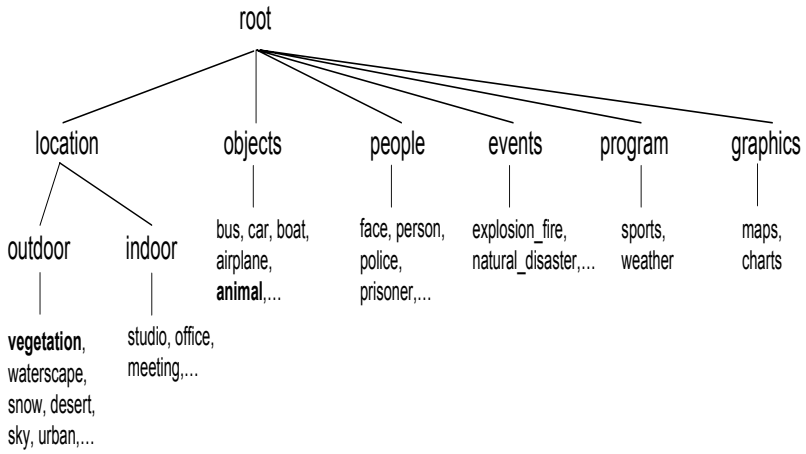


FIG. 4.5 – Fragment de l'ontologie hiérarchique LSCOM-Lite.

4.4.1 La cooccurrence

Elle est obtenue en considérant les statistiques d'apparence entre les concepts, où la présence ou l'absence de certains concepts peuvent prédire la présence ou l'absence d'autres concepts. Si on prend comme exemple les documents sur internet, les termes associés à des concepts qui co-occurrent souvent ont une plus grande probabilité d'être similaires dans un certain sens (e.g. les noms de sportifs qui se retrouvent sur la même page Web ont plus de chance d'appartenir à une discipline similaire). Il faut aussi tenir compte de la fréquence absolue d'occurrence des termes individuels (e.g. ZIDANE risque de se retrouver dans beaucoup plus de pages que d'autres joueurs). Proprement normalisées, ces fréquences de cooccurrence nous donnent donc de l'information sur la similarité entre concepts.

Plusieurs mesures peuvent être utiliser pour représenter cette information, on citera : la distance Minkowski (Euclidean, Manhattan), Hamming, Dice, similarité du cosinus,... Notre choix s'est porté sur cette dernière pour sa simplicité, l'exactitude des résultats par

rapport à d'autres mesures comme le produit scalaire et la distance Euclidienne [KSL07]. Cette mesure utilise la représentation vectorielle complète (i.e. la fréquence des concepts). Deux concepts sont similaires si leurs vecteurs sont confondus. Si deux concepts ne sont pas similaires, leurs vecteurs forment un angle θ dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs du concept C_m et C_n et le produit de la norme de C_m et de C_n.

$$Sim_{cos}(P^m, P^n) = \frac{\sum_{i=0}^{N-1} P_i^m P_i^n}{\sqrt{\sum_{i=0}^{N-1}(P_i^m)^2 \sum_{i=0}^{N-1}(P_i^n)^2}} \tag{4.6}$$

avec P^m est un vecteur binaire tel que :

$$P_i^m = \begin{cases} 1 & \text{si l'élément } i \in C_m \\ 0 & \text{ailleurs} \end{cases} \tag{4.7}$$

Le cosinus de l'angle entre deux vecteurs représentant la fréquence d'apparition entre deux concepts est égale à 1 si les deux concepts sont similaires, et égale à -1 s'ils sont entièrement différents.

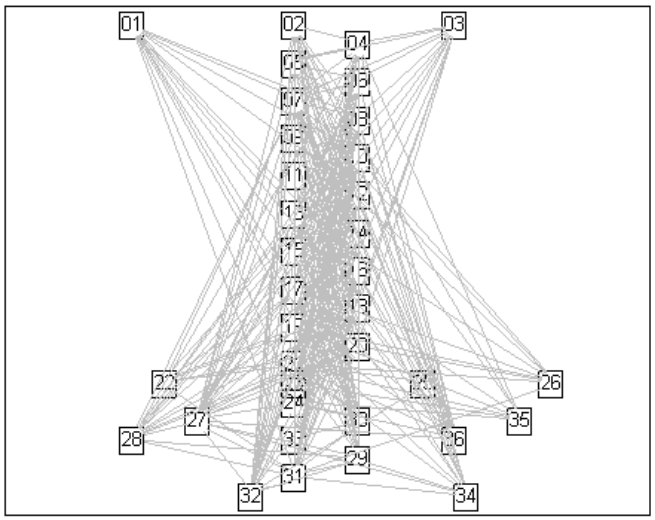


FIG. 4.6 – Représentation des connections inter-concepts, formant un modèle graphique. On peut associer une valeur pour chaque liaison, indiquant le degré de corrélation entre les deux concepts.

4.4.2 La similarité visuelle

La seconde similarité est basée sur les descripteurs visuels bas-niveau. Dans la section. 3.2.3, nous avons utilisé la notion d'entropie et de perplexité pour construire une relation de pondération entre les descripteurs et les concepts. Plusieurs mesures existent dont certaines sont des distances (i.e. des mesures dont les propriétés de non-négativité, réflexivité, symétrie et de l'inégalité triangulaire sont respectées) spécifiques aux histogrammes ou aux distributions. Dans [PBRT99], une étude comparative de 9 mesures est proposée. Nous nous sommes particulièrement intéressés par la similarité entre distributions, qui consiste à déterminer si deux distributions statistiques codées par un vecteur de caractéristiques peuvent être issues de la même distribution de probabilités. On citera [KSL07] : la *divergence de Kullback-Leibler, divergence de Jensen-Shannon, divergence de Jeffrey,* etc. C'est cette dernière qui sera utilisée dans nos travaux.

La divergence Kullback-Leibler est extraite de la théorie de l'information, elle mesure l'entropie relative de l'histogramme P^m par rapport à l'histogramme P^n (Equ. 4.8). La divergence est d'autant plus petite que les distributions sont proches. La divergence de KL est toujours positive, elle est nulle si et seulement si les deux distributions sont identiques. Il est important de noter que d_{KL} n'est pas une distance car elle ne respecte pas la propriété de symétrie.

$$d_{KL}(P^m, P^n) = \sum_{i=0}^{k-1} \left(P_i^m \log \frac{P_i^m}{P_i^n} \right) \tag{4.8}$$

avec P_i^m et P_i^n sont les probabilités de distributions des concepts C_m et C_n au tour des k centres obtenus précédemment dans la section 3.2.3.

La divergence de Jeffrey est une variante symétrique de d_{KL}. Elle est définie par :

$$d_{JD}(P^m, P^n) = \sum_{i=0}^{k-1} \left(P_i^m \log \frac{P_i^m}{\hat{P}_i} + P_i^n \log \frac{P_i^n}{\hat{P}_i} \right) \tag{4.9}$$

avec $\hat{P}_i = \frac{P^m + P^n}{2}$ est la moyenne de la distribution.

Enfin, la similarité visuelle globale entre deux concepts C_m et C_n, en utilisant l'ensemble des Nb descripteurs visuels bas-niveau est la suivante :

$$Sim_{vis}(C_m, C_n) = \frac{1}{\left(\sum_{i=1}^{Nb \ \ descripteurs} \frac{1}{2}(w_i^m + w_i^n) d_{JD}(P^{m,i}, P^{n,i}) \right)} \tag{4.10}$$

où w_i^m est le vecteur de pondération du $i^{\text{ème}}$ descripteur pour le calcul du PENN, par rapport au concept m.

4.4.3 La similarité sémantique

La similarité sémantique a été largement étudiée dans la littérature et dans beaucoup d'applications : résumé automatique, extraction d'information, indexation automatique, etc. Trois grandes approches se distinguent : (1) les approches basées sur la distance [RMBB89]

(i.e. sur la structure de l'ontologie), (2) les approches basées sur le contenu informatif des concepts [SC99, SVH04] et enfin (3) les approches mixtes ou hybrides qui combinent les deux approches précédentes [Res99].

4.4.3.1 Approche basée sur la distance

L'ontologie est représentée par un graphe ou les noeuds sont des concepts et les arcs sont les liens entre concepts. Cette approche considère que la similarité sémantique est calculée par le nombre d'arcs (liens) entre deux concepts. Selon le chemin suivi, plusieurs méthodes ont été proposées. Les inconvénients de cette approche sont sa dépendance à l'organisation des concepts dans la hiérarchie et que tous les liens (arcs) possèdent le même poids, ce qui impose des difficultés au niveau de la définition et du contrôle des distances des liens [SBM07]. On citera la mesure de *Edge Counting* de Rada et al. [RMBB89], qui utilise une distance $dist_{Rada}(C_m, C_n)$ indiquant le nombre d'arcs minimum séparant deux concepts (i.e. le plus court chemin dans la hiérarchie), comme le montre l'équation 4.11. Plus deux concepts sont distants, moins ils sont similaires.

$$Sim_{sem}(C_m, C_n) = \frac{1}{1 + dist_{Rada}(C_m, C_n)} \tag{4.11}$$

Wu et Palmer [WP94] proposent une similarité basée sur le plus petit généralisant commun obtenu par le calcul de la profondeur du concept subsumant CS [8] $depth(CS)$ et des deux concepts, comme le montre l'exemple dans la Fig. 4.7. Cette mesure est intéressante mais présente une limite car elle vise essentiellement à détecter la similarité entre deux concepts par rapport à leur distance du concept CS. Plus ce dernier est général, moins les deux concepts sont similaires (et inversement).

$$Sim_{sem}(C_m, C_n) = \frac{2 * depth(CS)}{depth(C_m) + depth(C_n)} \tag{4.12}$$

D'autres travaux utilisent le chemin le plus long de la hiérarchie D à l'image des travaux de Fan et al. [FGL07] par l'équation 4.13.

$$Sim_{sem}(C_m, C_n) = -\log\left(\frac{dist_{Rada}(C_m, C_n)}{2D}\right) \tag{4.13}$$

4.4.3.2 Approche basée sur le contenu informatif des noeuds

Cette approche prend en compte l'information partagée par les concepts en termes de mesure entropique de la théorie de l'information. Deux méthodes existent. La première utilise un corpus d'apprentissage et calcul la probabilité $p(C_i)$ de trouver un concept C_i ou un de ses descendants dans ce corpus. Selon Resnik [Res95] la similarité sémantique entre deux concepts est mesurée par la quantité de l'information qu'ils partagent, en calculant la fréquence d'apparition dans le corpus. La formule proposée est définie par :

$$Sim_{sem}(C_m, C_n) = \max\left(IC(CS(C_m, C_n))\right) \tag{4.14}$$

[8] Le concept subsumant c'est le concept commun le plus spécifique.

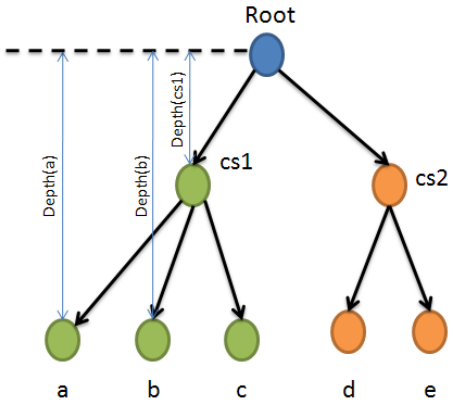


FIG. 4.7 – Exemple d'un extrait d'ontologie pour le calcul de la similarité de Wu et Palmer [WP94]. La similarité sémantique entre le concept (a) et (b) revient à calculer la profondeur du concept subsumant (CS1) sur la somme de la profondeur des deux concepts par rapport à la racine.

avec $IC(C_i) = -\log(p(C_i))$ est le contenu de l'information d'un concept C_i (i.e. l'entropie d'une classe C_i). La probabilité $p(C_i)$ est calculée en divisant le nombre des instances de C_i par le nombre total des instances. $CS(C_m, C_n)$ représente le concept le plus spécifique (qui maximise la valeur de similarité) qui subsume (situé à un niveau hiérarchique plus élevé) les deux concepts C_m et C_n dans l'ontologie. Cette mesure est un peu sommaire car elle ne dépend que du concept subsumant CS.

La deuxième méthode calcule le contenu informatif des noeuds à partir de WordNet au lieu d'un corpus [Fel98]. Seco et al. [SVH04] utilisent les hyponymes [9] descendants des concepts pour calculer le contenu informatif de ceux ci.

$$IC_{wn}(C) = \frac{\log\left(\frac{hypo(C)+1}{max_{wn}}\right)}{\log\left(\frac{1}{max_{wn}}\right)} = 1 - \frac{\log\left(hypo(C)+1\right)}{\log\left(max_{wn}\right)} \tag{4.15}$$

avec $hypo(C)$ est le nombre d'hyponymes du concepts C et max_{wn} est une constante qui indique le nombre de concepts de la taxonomie.

Cette approche peut obtenir une valeur de similarité de deux éléments d'une ontologie contenus dans le voisinage qui dépasse la valeur de similarité de deux concepts contenus dans la même hiérarchie. Cette situation est inadéquate dans le cadre de la recherche de l'information [SBM07].

[9] La relation sémantique de base entre les mots codés dans WordNet est la synonymie. Les synsets sont liés par les relations d'hypernymies, d'hyponymies, d'holonymies et de meronymies, etc [LC98].
- La synonymie représente un ensemble de mots qui sont interchangeables dans un contexte donné.
- L'hypernymie désigne une classe englobant des instances de classes plus spécifiques. Y est un hypernyme de X si X est un type de Y.
- L'hyponymie désigne un membre d'une classe. X est un hyponyme de Y si X est un type de Y.

4.4.3.3 Approche mixte/hybride

Les deux grandes approches définies précédemment peuvent être combinées. Souvent, il s'agit de réutiliser le contenu informatif des noeuds et le plus petit ancêtre commun, comme avec la méthode proposée par Lin. [Lin98] :

$$Sim_{sem}(C_m, C_n) = \frac{2 * \log\left(P(CS)\right)}{\log\left(P(C_m)\right) + \log\left(P(C_n)\right)} \tag{4.16}$$

ou encore avec la distance de Jiang & Conrath $dist_{J\&C}$ [JC97].

$$Sim_{sem}(C_m, C_n) = \frac{1}{dist_{J\&C}(C_m, C_n)} = \frac{1}{IC_{\text{Resnik}}(C_m) + IC_{\text{Resnik}}(C_n) - 2 * IC_{\text{Resnik}}(CS(C_m, C_n))} \tag{4.17}$$

4.4.3.4 Notre démarche

Notre choix part d'une phase d'observation sur l'ontologie LSCOM-lite représentée dans la Fig. 4.8. Il est clair qu'elle ressemble plus à un arbre hiérarchique qu'à une ontologie comme WordNet. Néanmoins, elle a permis un regroupement de concepts selon un certain sens. Pour cela, nous pensons que l'utilisation de cette dernière, ne pourra qu'être bénéfique malgré ses limitations. Le choix de la méthode de calcul de la similarité sémantique se portera d'abord sur les deux approches hybrides de Jiang & Conrath [JC97] et de Lin [Lin98] précédemment citées. Ces deux approches présentent l'avantage d'être simple à calculer en plus des performances qu'elles réalisent. Ensuite, nous étendrons à une nouvelle mesure hybride combinant la distance de Rada [RMBB89] avec celle de Jiang & Conrath afin de ne pas dépendre seulement du contenu informatif des noeuds, en particulier du concept subsumant, et de rajouter la distance issue du graphe entre les deux concepts. On notera l'équation par B&H.

$$Sim_{sem}(C_m, C_n) = \frac{1}{dist_{Rada}(C_m, C_n) + dist_{J\&C}(C_m, C_n)} \tag{4.18}$$

4.5 Réajustement des valeurs de confidence basé sur la similarité inter-concepts

Notre schéma proposé dans la Fig. 4.4 introduit une mise à jour ou un réajustement *"reranking"* des valeurs de confidence obtenues par le PENN préalablement utilisé, pour raffiner les résultats de la détection à travers l'équation suivante :

$$\underline{P\left(x/Ci\right)} = P\left(x/Ci\right) + \frac{1}{Z} \sum_{j=1}^{Nb\ arc} \lambda_{i,j}(1 - \zeta_j) P\left(x/Cj\right) \tag{4.19}$$

avec $\underline{P\left(x/Ci\right)}$ est le résultat multimodal (sortie du PENN), $\lambda_{i,j}$ est le degré de similarité entre le concept C_i et C_j, ζ_j est l'erreur de classification obtenue pour l'ensemble de validation (i.e. cette erreur est introduite pour pondérer la réponse des noeuds voisins). Z est un terme de normalisation.

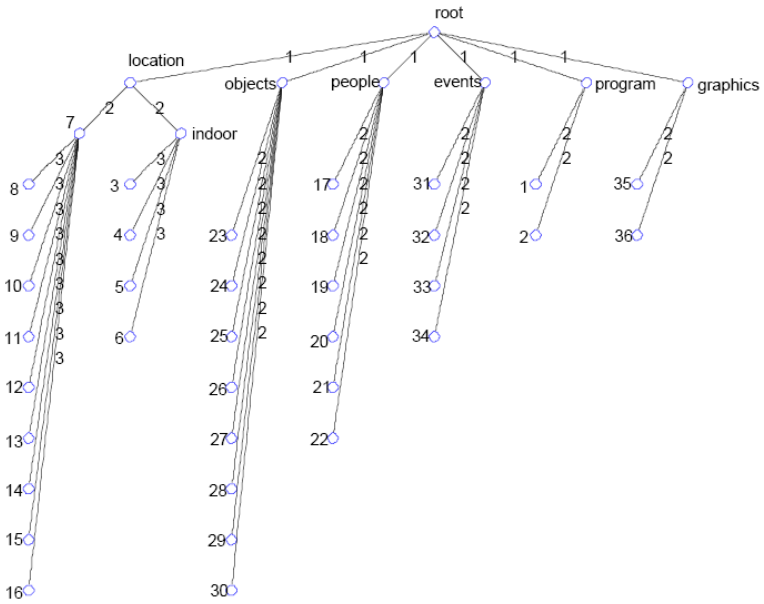


FIG. 4.8 – Modèle d'ontologie hiérarchique. Le *root* est la racine de l'ontologie, les noeuds représentent les différents concepts de 1 à 36 (voir la Table 1.5) regroupés dans 6 catégories (programmes, localisation, objets, événements, etc). Chaque arc est lié à une valeur de profondeur par rapport à la racine.

4.6 Evaluation

La Fig. 4.9 présente le comportement des trois systèmes d'indexations NNET, PENN et Onto-PENN [10], en termes de la précision moyenne par concept. Premièrement, nous observons que les systèmes PENN et Onto-PENN produisent des performances identiques pour plusieurs concepts, avec une amélioration significative comparant à NNET (voir 4,6,17,18,19, 23,31 et 32). Ce résultat n'est pas une surprise mais due à la manière de calculer la précision moyenne AP. Celle-ci suit le protocole d'évaluation de TRECVid obtenu pour les 2000 premiers plans retournés [TRE]. Deuxièmement, quelques faibles performances peuvent être observées à cause des situations de conflits et des limitations de l'ensemble d'entraînement. Ceci explique les cas extrêmes obtenus pour les concepts 3,22,25,26,33 et 34.

Pour évaluer la contribution de la similarité inter-concepts sur le système d'indexation, nous allons étudier les résultats sur la totalité de l'ensemble du test. Pour cela, la comparaison des performances sera liée à un seuil [11] de classification choisi à travers F-meas, CR^+ et BER. Notons que AP n'est pas sensible à ce seuil.

Les Figures 4.10 à 4.12 comparent les trois systèmes expérimentaux en fonction de la variation du seuil $\in [0.1, 0.9]$ par un pas de 0.1. On observe clairement que quelque soit le

[10] Les résultats présentés pour Onto-PENN sont ceux obtenus par l'application de la similarité sémantique avec l'équation 4.18.

[11] **Si** (soft-decision $\geq$ seuil $\in [0.1, 0.9]$), **alors** le plan est classé comme *Class*, **sinon** comme *Non-Class*.

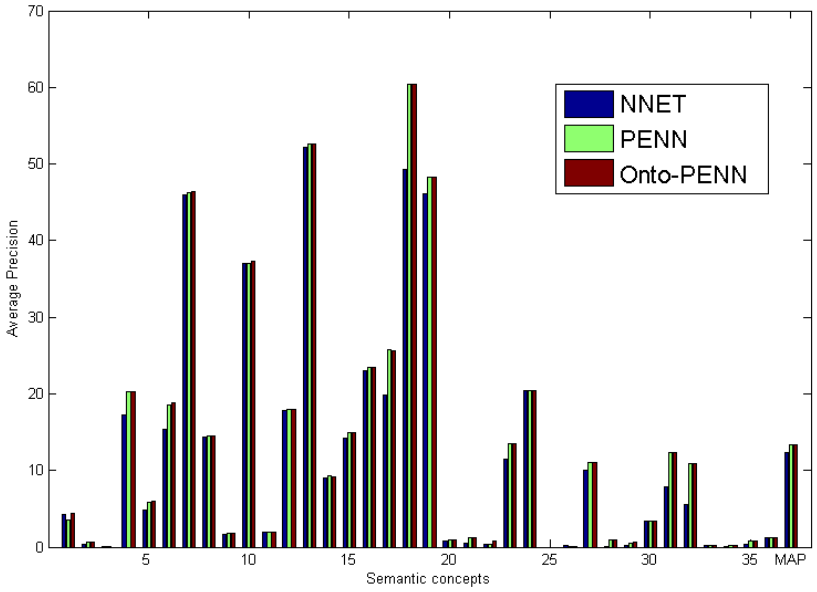


FIG. 4.9 – Evaluation des systèmes par concepts, en utilisant la précision moyenne.

seuil choisi Onto-PENN domine et obtient des performances élevées pour F-meas, CR^+ et diminue l'erreur BER par comparaison à PENN et NNET. Le $BER_{min} = 40.38\%$ est donné par le seuil= 0.2, avec F-meas= 16.98% et $CR^+ = 34.48\%$.

Les meilleurs résultats sont obtenus dans l'intervalle de seuil $[0.2, 0.5]$. En effet, pour un seuil fixé à 0.40, on rapporte un gain de 10.14% pour atteindre le $CR^+ = 22.07\%$, reduisant l'erreur BER de 2.91% comparant à NNET.

Les Figures. 4.13 et 4.14 présentent l'évolution des performances pour les trois systèmes par concepts, en utilisant un seuil= 0.4. Plusieurs points peuvent être notés, comme suit :

- Les trois systèmes produisent une certaine non-détection (F-meas= 0, $CR^+ = 0$) pour les concepts 2,3,9,11,25,26,28,29,33,34 et 36.
- En plus de cela, NNET ne détecte aucun des concepts 1,5,6,20,21,22,31,32 et 35. De même pour PENN sur les concepts 5,20,22 et 35.
- Onto-PENN résout la dernière limite et produit une importante amélioration (voir Fig. 4.14) pour les concepts 1,4,5,7,8,10,12,13,15,16,17,18,19,21,22,23,24,27 et 31, due à l'apport positif de la similarité inter-concepts dans la mise à jour conduisant à une meilleure prise de décision.

Par exemple, pour la détection des concepts FACE, PERSON, MEETING ou STUDIO, le système PENN attribue plus d'importance aux descripteurs *FaceDetector, ContourShape, ColorLayout, ScalableColor, EdgeHistogram* qu'aux autres. Pour le concept "FACE", l'amélioration est de 11%, la placant comme notre plus grande performance. Par consequent, Onto-PENN introduit les relations sémantiques entre les concepts connectés, améliorant la qualité des performances (voir Fig. 4.15).

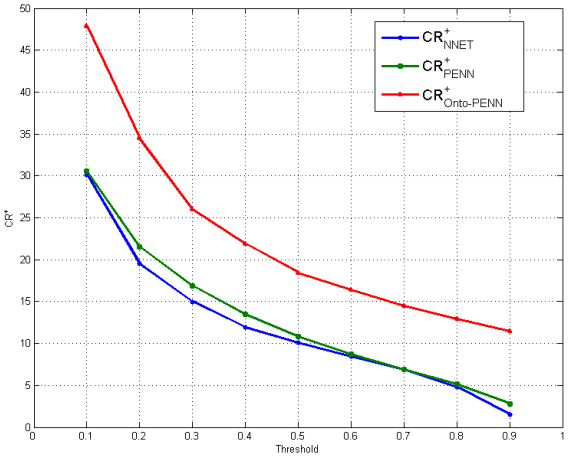


FIG. 4.10 – Evolution du CR^+ *vs* seuil $\in [0.1, 0.9]$.

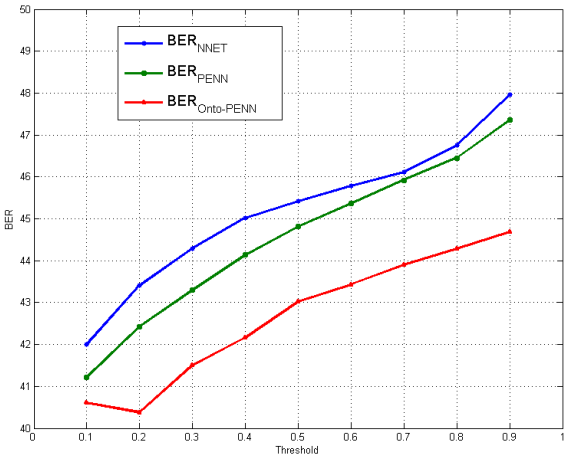


FIG. 4.11 – Evolution de l'erreur BER *vs* seuil $\in [0.1, 0.9]$.

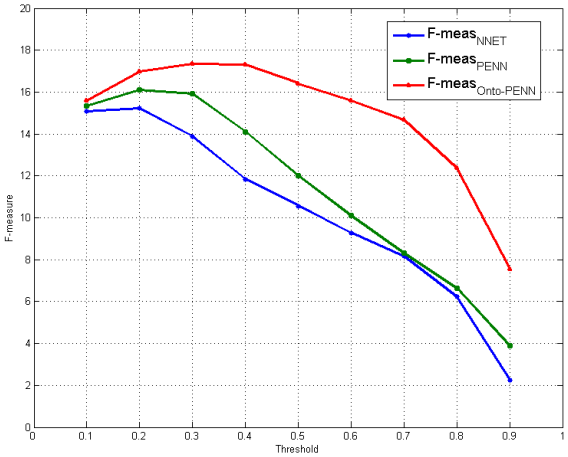


FIG. 4.12 – Evolution de la F-mesure *vs* seuil $\in [0.1, 0.9]$.

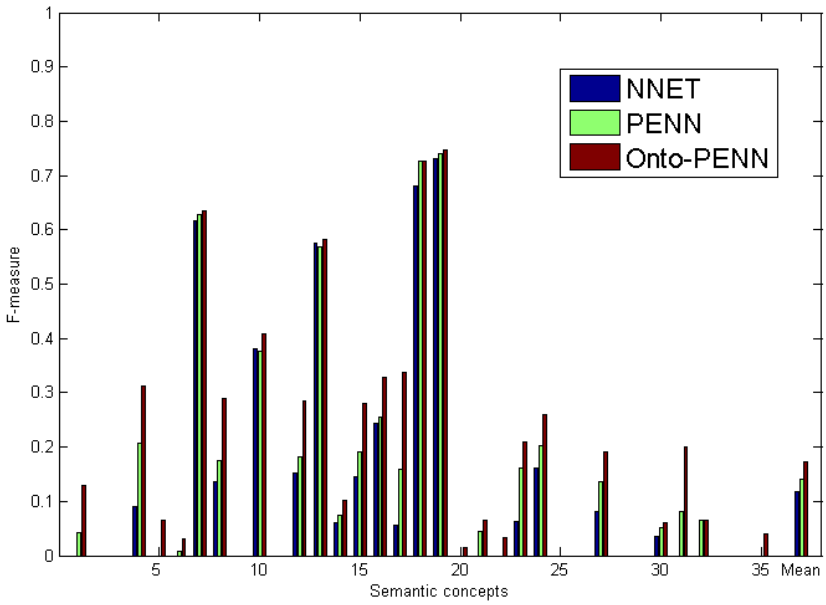


FIG. 4.13 – Evaluation des systèmes par concepts, en utilisant la F-mesure.

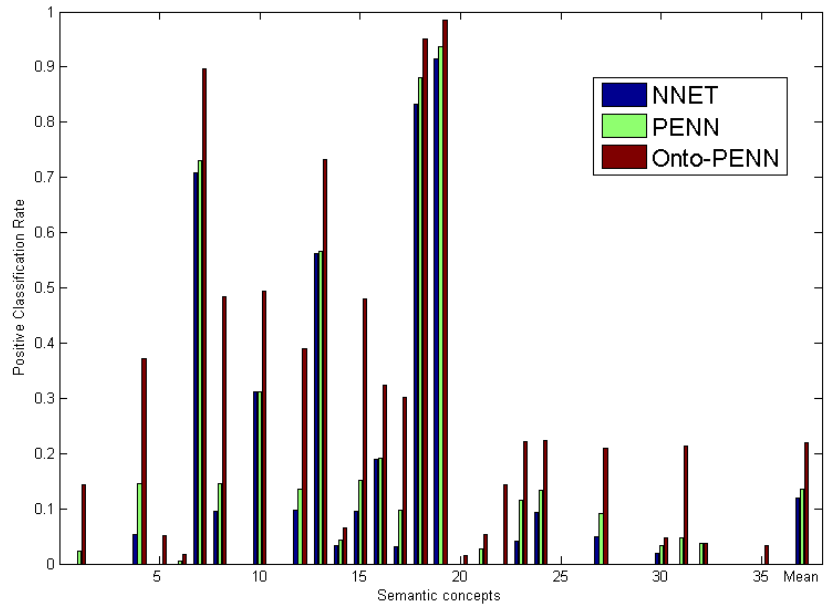


FIG. 4.14 – Evaluation des systèmes par concepts, en utilisant le CR^+.

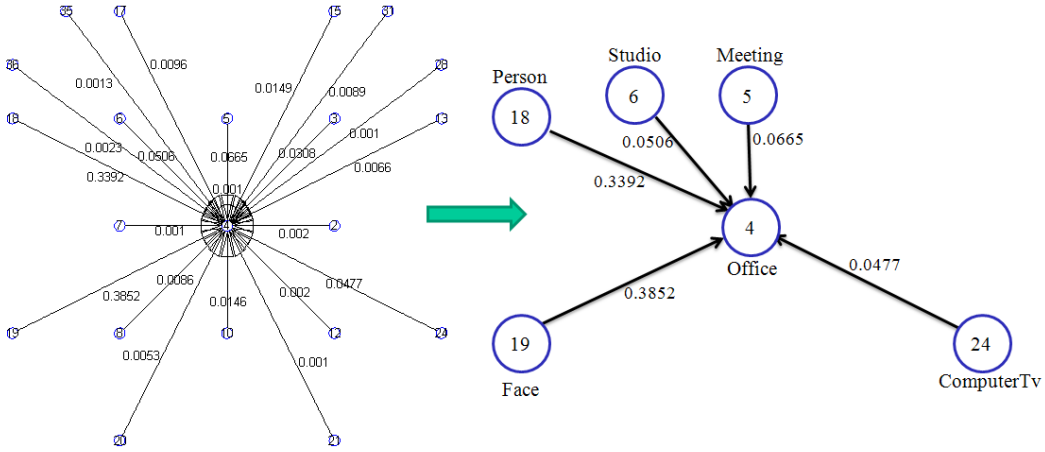


FIG. 4.15 – Représentation des liaisons inter-concepts pour le noeud central OFFICE, associant une valeur pour chaque liaison, qui indique le degré de corrélation des deux concepts. On observe que 20 concepts ont des connections avec OFFICE, mais juste les 5 suivantes sont fortes et signifiantes : MEETING :6.65%, STUDIO :5.06%, FACE :33.92%, PERSON :38.52% et COMPUTERTV :4.77%, présentant 88.92% de l'information globale.

La Table 4.1 résume l'ensemble des performances pour les trois systèmes proposés en utilisant un seuil= 0.4. On détermine les statistiques précédentes pour tous les concepts et pour l'ensemble des 10 concepts les plus fréquents dans la base. Onto-PENN offre plusieurs améliorations sur le F-meas et CR^+, avec un résultat respectable pour le MAP et une diminution de l'erreur "BER" par comparaison aux systèmes NNET et PENN.

Enfin, les équations utilisées dans la construction de la similarité sémantique hybrides expriment des résultats très proches, avec un avantage pour la similarité B&H (Equ. 4.18). Par ailleurs, on remarque une baisse de la précision moyenne lors de l'utilisation de l'approche basée sur la distance proposée par Rada (Equ. 4.11), et hybride de Lin et al (Equ. 4.16), mais elle reste largement supérieure à celle du NNET. Ceci nous montre l'importance du choix de la méthode de modélisation de la similarité sémantique, en particulier pour ce genre d'ontologie.

4.7 Conclusion

Dans ce chapitre, nous avons proposé le système Onto-PENN qui permet l'exploitation de l'information sémantique et à priori par les systèmes de fusion. Trois types d'influences (de relations) sont utilisés : la cooccurrence, les descripteurs visuels bas-niveau et la similarité sémantique via l'approche hybride, pour l'amélioration de la précision du système conceptuel. Seule la meilleure approche est conservée, basée sur la combinaison entre l'approche de Rada [RMBB89] et celle de Jiang & Conrath [JC97] dans l'ontologie LSCOM-lite, adaptable à l'analyse conjointe de plusieurs classes. Les résultats obtenus présentent des améliorations

Methods / Eval.(%)	NNET	PENN	Onto-PENN			
			Rada	Lin	J&C	B&H
MAP	12.70	13.29	12.94	13.01	13.31	13.37
MAP@10	33.70	35.30	34.12	34.91	35.30	35.36
F-meas	11.84	14.10	15.97	16.17	17.07	17.30
F-meas@10	38.75	40.79	41.83	43.41	44.67	44.74
CR^+	11.93	13.43	18.12	20.58	21.76	22.07
CR^+@10	40.69	41.74	53.76	57.80	59.45	59.71
BER	45.02	44.13	43.93	43.62	42.32	42.11
BER@10	38	36.52	36.02	35.45	34.03	33.96

TAB. 4.1 – Comparaison des performances entre les trois systèmes expérimentaux NNET, PENN et Onto-PENN. Aussi, on montre l'effet de l'utilisation des approches suivantes : Rada (Equ. 4.11), Lin (Equ. 4.16), J&C (Equ. 4.17) avec celle proposée B&H (Equ. 4.18) sur le système Onto-PENN, pour un *seuil*= 0.4.

importantes de 18.75% pour CR^+, 5.99% pour F-mesure, 1.66% pour MAP et enfin diminue l'erreur BER de 2.91%.

Il est à signaler que le corpus et l'ontologie utilisée par sa taille et sa forme de regroupement des concepts n'est peut être pas la meilleure. Ainsi, une intégration d'une solution telle que WordNet pourrait améliorer la généralisation de l'algorithme.

Conclusions et Perspectives

Dans ce mémoire, nous nous sommes intéressés à l'indexation et à la recherche des plans vidéo par le contenu. Plusieurs problèmes ont été soulevés, notamment, le rôle de la fusion multi-niveaux dans la réduction du fossé sémantique entre les descripteurs bas-niveau et les concepts sémantiques. Les solutions proposées ont été évaluées dans le cadre de deux projets de recherche : CRE Fusion et NoE K-Space. Les résultats d'évaluations nous ont confirmé l'intérêt de la fusion dans le système d'indexation proposé.

Résumé

Le développement et l'accès aux bases de données nécessitent de structurer l'information du contenu. Aujourd'hui, la recherche et la classification d'images ou des vidéos dans les grandes bases de données, d'archives et sur internet s'effectuent principalement grâce à des données textuelles : nom de l'image, mots-clés... Cependant, cette indexation est extrêmement coûteuse et non exempte de fautes plus ou moins graves : omission, orthographe, etc. Les progrès effectués sur l'analyse d'images et l'apprentissage automatique permettent d'apporter des solutions comme l'indexation et la recherche à base des caractéristiques telles que la couleur, la forme, la texture, le mouvement, le son et le texte. Ces caractéristiques sont riches en informations et notamment d'un point de vue sémantique. Cependant, les méthodes proposées sont semi-automatiques (retour d'informations par l'utilisateur) ou automatiques après une phase importante d'apprentissage sur une base de petite ou moyenne taille. La fusion est utilisée dans le but de combiner des informations issues de plusieurs sources pour obtenir une information globale plus complète, de meilleure qualité et permettant de mieux décider et d'agir.

En effet, la fusion peut être effectuée dans différents niveaux du système de classification. Nous avons étudié la fusion de descripteurs bas-niveau (fusion précoce) *"feature fusion"*, la fusion de descripteurs haut-niveau (fusion tardive) *"classifier fusion"* et à un niveau plus élevé pour la prise de décision finale à base de l'ontologie et de la similarité inter-concepts *"decision fusion"*. Dans cette thèse, nous avons présenté un tour d'horizon sur les travaux effectués dans le domaine de la fusion et les outils permettant de les exploiter dans le système d'indexation, par des aspects scientifiques et techniques.

Au cours de ce travail, nous avons développé un modèle multi-niveaux de construction et d'utilisation de la classification multimédia, appliqué à l'indexation des plans vidéo TRECVid. D'abord, nous avons montré que le modèle de fusion haut-niveau, en particulier

Methods / Eval.(%)	NNET	PENN	Onto-PENN
MAP	12.70	13.29	13.37
MAP@10	33.70	35.30	35.36
F-meas	11.84	14.10	17.30
F-meas@10	38.75	40.79	44.74
CR^+	11.93	13.43	22.07
CR^+@10	40.69	41.74	59.71
BER	45.02	44.13	42.11
BER@10	38	36.52	33.96

TAB. 4.2 – Résumé des performances entre les trois systèmes expérimentaux NNET, PENN et Onto-PENN.

par l'introduction de la théorie des évidences à travers le NNET (Neural Network based on Evidence Theory) permet de déterminer avec une performance satisfaisante plusieurs concepts présents dans les plans, dans la mesure où les descripteurs choisis sont pertinents pour la classification. Puis, l'application de la méthode PENN (Perplexity-based Evidential Neural Network) via deux étapes : (1) la pondération des sorties de classifieurs à travers l'étude de la perplexité, qui estime le degré de relation entre les descripteurs et les concepts, (2) et la combinaison NNET. Par ailleurs, l'introduction de l'ontologie par la similarité inter-concepts Onto-PENN a permis l'exploitation de l'information sémantique et à priori par notre système. Trois types d'influences (de relations) ont été utilisés : la cooccurrence, les descripteurs visuels bas-niveau et la similarité sémantique via l'approche hybride, pour l'amélioration de la précision du système conceptuel. Seule la meilleure approche est conservée, basée sur la combinaison entre l'équation de Rada et celle de Jiang & Conrath dans l'ontologie LSCOM-lite, adaptable à l'analyse conjointe de plusieurs classes. Les résultats obtenus présentent des améliorations importantes de 18.75% pour CR^+, 5.99% pour F-mesure, 1.66% pour MAP et enfin, diminue l'erreur BER de 2.91% comme le montre la Table. 4.2.

D'autre part, nous avons constaté une augmentation des performances sur la base de vidéos de sport, en combinant l'information issue des descripteurs au sein d'un modèle non-linéaire de réduction de la dimension NNC (Neural Network Coder). En effet, la fusion joue un rôle primordial dans la classification, et semble être plus efficace en amont des systèmes de classifications qu'en aval. Une limitation concerne nos conditions de test de l'algorithme sur les vidéos de football pour en pouvoir généraliser. Dans ces circonstances, nous l'avons appliqué sur les données TRECVid. Les propriétés de la méthode sont mises à profit pour avoir une meilleure réduction de la dimension, tout en soulignant la forte corrélation entre la quantité des données et les performances du système. Cependant, cette tache est longue et fastidieuse mais ouvre une porte et des perspectives pour ce champs de recherche.

Enfin, d'autres expérimentations avec le NNET ont été réalisées. La première concerne l'effet de la fusion sur la détection des émotions humaines [PBH09] (ANGER, DISGUST,

FEAR, HAPPINESS, SADNESS, SURPRISE), la deuxième sur l'indexation et l'annotation des régions d'images [ASP+09], avec de meilleures performances que les méthodes numériques existantes.

Perspectives

Plusieurs directions peuvent être envisagées dans la continuité de nos travaux. Nous retiendrons en particulier les points suivants :

- L'enrichissement de la base de descripteurs par des informations issues de l'audio et du texte, pour une meilleure capture du contenu. Les récentes campagnes d'évaluations TRECVid ont démontré cela. Plusieurs équipes utilisent à présent de nombreux descripteurs provoquant une forte concurrence entre les participants. Ainsi, dans le cadre de l'évaluation TRECVid'07 [BGH07], cinq systèmes (*runs*) ont été soumis dont un (*run 4*) intègre le descripteur audio MFCC (i.e. Mel-frequency cepstral coefficients). Les résultats obtenus montrent des améliorations sur le taux de reconnaissance. D'autres dérivations des coefficients cepstraux peuvent être utilisées (e.g. LPCC "Linear Prediction cepstral coefficients", PLP-CC "Perceptual Linear Prediction cepstral", etc) [Her90, HSHB05]. Cependant, nous devrons veiller à ce que le nombre de dimensions ne soit pas trop important, en associant des algorithmes de réduction de dimension pour ne conserver que celles porteuses d'informations. Nous devons également prendre en compte le problème inhérent à la désynchronisation du contenu visuel et du contenu audio et textuel. La fusion pourrait atténuer ce problème en modélisant le décalage entre les différentes modalités.

- Parmi les autres soumissions, le *run 3*, qui représente le produit de la fusion des descripteurs MPEG-7 locaux uniquement. Ses résultats étaient très faibles en comparaison avec le *run 1* issu des descripteurs globaux. Cette surprise, nous a révélé le mauvais choix du nombre de mots-clés visuels pour la quantification (NbCluster=50). En effet, le choix de ce nombre est souvent arbitraire ou obtenu par l'observation des performances moyennes. Cette dégradation de performance provient du faible nombre de mots-clés visuels choisis lors de l'apprentissage, décrivant ainsi le contenu par une palette très pauvre où tout semble identique. Afin d'étudier l'impact de ce nombre sur les performances du système, nous avons repris les expériences (avec un test différent) en faisant varier ce nombre entre 100 et 2000, comme le montre la Fig. 4.16. Plus le nombre de mots-clés augmente plus le MAP s'améliore, où avec un $NbCluster = 500$, on accroît les résultats obtenus par les descripteurs globaux. Cependant, le nombre optimal pour décrire le contenu visuel est étroitement lié aux requêtes. Le choix ne sera donc pas idéal pour toutes les requêtes envisageables mais en moyenne, les performances sont stables sur l'intervalle $[500, 2000]$ qui est assez large. Tout en faisant attention de ne pas choisir beaucoup de mots-clés visuels, décrivant ainsi le contenu par une palette très riche ou tout semble différent. Par ailleurs, la relation entre le mot-clé visuel et sa position dans l'image peut être aussi prise en compte dans cette étude.

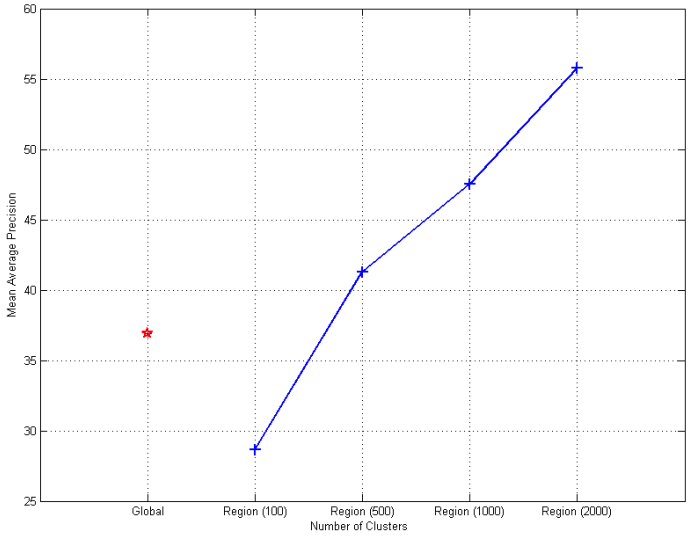


FIG. 4.16 – Choix du nombre de mots-clés visuels pour la quantification pour les descripteurs MPEG-7 locaux.

– Il est plus que nécessaire d'accélérer les traitements de notre système en réduisant sa complexité afin de pouvoir dans la suite nous confronter à des corpus plus importants. Dans nos travaux, nous avons utilisé un ensemble de 100 heures de vidéos, soit environ 60 Giga-octets, ce qui peut être considéré comme un grand corpus. Cependant, l'explosion du partage de vidéos sur internet, à l'image de YouTube [12], Dailymotion, Wimeo, MySpace, etc, et dans un esprit de réalisme. Le besoin de contrôler le coût des algorithmes est primordial. Par conséquent, nous pensons qu'une sélection pertinente du nombre d'exemples d'apprentissage, donc des vecteurs supports, est un bon moyen d'accélérer le temps de la classification SVM.

– Nous avons essentiellement abordé le thème important des ontologies à travers la similarité inter-concepts (i.e. l'étude des relations entre les classes). En effet, les concepts ne sont pas exprimés de manière isolée et une forte corrélation existe entre certaines classes. Pour cela, une étude de la similarité inter-concepts a été effectuée en introduisant 3 types d'informations : les descripteurs bas-niveau, la cooccurrence et la similarité sémantique issue de l'approche hybride. Cette étude ouvre des perspectives intéressantes car elle offre une passerelle entre une description de bas-niveau du plan

[12] La domination du moteur de recherche Google, propriétaire de YouTube (le plus grand site de publication de vidéos) n'est pas une surprise. Il a ainsi fourni 45% des lectures de vidéos en ligne au Royaume-Uni, 44% au Canada, 38% en Allemagne, 34% aux États-Unis et 28% en France. La faible proportion en France est notamment due à la bonne résistance du site de vidéos local Dailymotion, qui a diffusé 15,5% des vidéos regardées dans l'hexagone.

par des descripteurs visuels et une description de haut-niveau. Nous pensons que le corpus et l'ontologie utilisée par sa taille et sa forme de regroupement des concepts ne sont peut-être pas les meilleures. Une intégration d'une solution comme WordNet pourrait améliorer la généralisation de l'algorithme. Par ailleurs, une étude de la boucle d'asservissement permettra au système d'affiner ses réponses en utilisant un retour d'information de l'utilisateur sur le dernier résultat de la recherche.

– Concernant l'indexation des images-clés issues des vidéos de football, il est envisageable de poursuivre quelques travaux obtenus dans le cadre d'un projet étudiant interne que nous avons encadré, au sein d'Eurécom. En effet, l'intégration des points d'intérêts dans le système pourra être d'un grand apport. Trois détecteurs de points d'intérêts ont été étudiés et évalués : Harris, Hessian et SIFT (Scale-invariant feature transform). Les premiers résultats montrent que le SIFT est une technique efficace pour détecter des objets qui se trouvent dans l'image, même s'ils ont subi des traitements de différentes sortes. Cependant, dans notre application, trouver les objets spécifiques (par exemple, les joueurs) n'est pas suffisant pour déterminer la catégorie à laquelle appartient cette image. La variation de ces objets peut être critique pour la classification. Par exemple, les joueurs dans les images de la classe Zoom on Player sont généralement plus grands que ceux dans les images de la classe Close-up action. La Fig. 4.17 présente une image appartenant à la classe Close-up action, reconnue comme Goal camera, on peut y voir dans l'arrière plan les filets du but si caractéristiques. Dans ces deux cas, la caractéristique unique de la technique SIFT (i.e. invariance à l'échelle) est complètement défavorable.
Par ailleurs, le taux de reconnaissance par classes est aussi influencé par le nombre d'images-clés positives dans la base d'entraînement. L'intégration de nouvelles vidéos issues de différents matchs et de leur vérité terrain ne pourra qu'augmenter les chances d'une image test d'être bien reconnue.

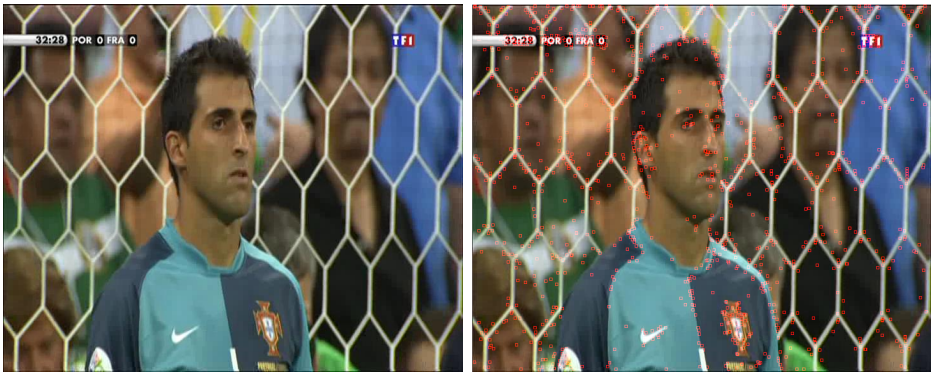


Fig. 4.17 – Exemple d'erreur obtenue pour une image appartenant à la classe Close-up action, reconnue comme Goal camera due à invariance à l'échelle du détecteur SIFT.

Annexe

Extraits de la base TRECVid 2007

Plans en relation avec la classe SPORTS ($Id = 1$)

Plans en relation avec la classe BUILDING ($Id = 8$)

Plans en relation avec la classe VEGETATION ($Id = 10$)

Plans en relation avec la classe WATERSCAPE ($Id = 16$)

Plans en relation avec la classe FACE ($Id = 18$)

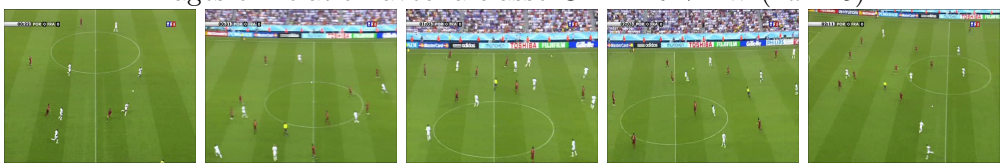
Plans en relation avec la classe CAR ($Id = 27$)

Plans en relation avec la classe MAPS ($Id = 35$)

Extraits de la base d'Orange-France Télécom Labs

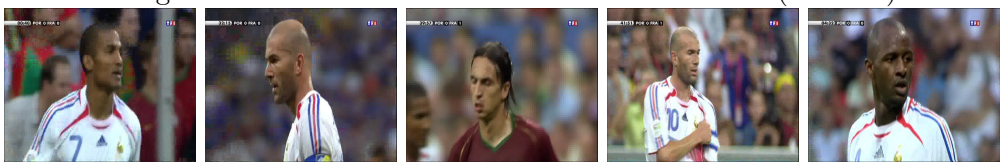
Images en relation avec la classe CLOSE-UP ACTION ($Id = 1$)

Images en relation avec la classe CENTER VIEW ($Id = 5$)



Images en relation avec la classe LEFT VIEW ($Id = 8$)

Images en relation avec la classe ZOOM ON PLAYER ($Id = 10$)



Nos Publications

Chapitre de Livre

1. R. Benmokhtar, B. Huet, G. Richard, S. Essid. Feature Extraction For Multimedia Analysis, Chapter in "Advances in Multimedia Semantics", Wiley, 2009.

Conférences avec comité de sélection

1. R. Benmokhtar, B. Huet. Classifier fusion : Combination methods for semantic indexing in video content, *Proceedings of ICANN*, volume 2, pages 65-74, Athens-Greece, September 2006.
2. R. Benmokhtar, B. Huet. Neural network classifier fusion based on Dempster-Shafer theory for semantic indexing in video content, *Proceedings of MMM*, volume 1, pages 196-205, Singapore, January 2007.
3. R. Benmokhtar, B. Huet. Multi-level fusion for semantic indexing video content, *Proceedings of AMR*, pages 160-169, Paris-France, June 2007.
4. R. Benmokhtar, B. Huet, S-A. Berrani, P. Lechat. Video shots key-frames indexing and retrieval through pattern analysis and fusion techniques, *Proceedings of FUSION*, pages 1-6, Quebec-Canada, July 2007.
5. R. Benmokhtar, B. Huet, S-A. Berrani. Low-level feature fusion models for soccer scenes classification, *Proceedings of IEEE ICME*, pages 1329-1332, Hanover-Germany, June 2008.
6. R. Benmokhtar, B. Huet. Perplexity-based evidential neural network classifier fusion using MPEG-7 low-level visual features, *Proceedings of ACM MIR*, pages 160-169, Vancouver-Canada, October 2008.
7. M. Paleari, R. Benmokhtar, B. Huet. Evidence theory-based multimodal emotion recognition, *Proceedings of MMM*, pages 435-446, Nice-France, January 2009.
8. T. Athanasiadis, N. Simou, G. Papadopoulos, R. Benmokhtar, K. Chandramouli, V. Tzouvaras, V. Mezaris, M. Phiniketos, Y. Avrithis, I. Kompatsiaris, B. Huet, E. Izquierdo. Integrating image segmentation and classification for fuzzy knowledge-based multimedia indexing, *Proceedings of MMM*, pages 263-274, Nice-France, January 2009.
9. R. Benmokhtar, B. Huet. Ontological reranking approach for hybrid concept similarity-based video shots indexing, *Proceedings of WIAMIS*, London-UK, May 2009.
10. R. Benmokhtar, B. Huet. Hierarchical ontology-based robust video shots indexing using global MPEG-7 visual descriptors, *Proceedings of CBMI*, Crete Island-Greece, June 2009.

Articles publiés sans comité de sélection

1. R. Benmokhtar, E. Dumont, B. Huet, B. Merialdo. Eurécom at TRECVid 2006 : Extraction of high level features and BBC rushes exploitation, *Proceedings of TREC*, Gaithersburg-USA, November 2006.
2. R. Benmokhtar, E. Dumont, B. Huet, B. Merialdo. K-Space at TRECVid 2006, *Proceedings of TREC*, Gaithersburg-USA, November 2006.
3. R. Benmokhtar, E. Galmar, B. Huet. Eurécom at TRECVid 2007 : High level features extraction, *Proceedings of TREC*, Gaithersburg-USA, November 2007.
4. R. Benmokhtar, E. Galmar, B. Huet. K-Space at TRECVid 2007, *Proceedings of TREC*, Gaithersburg-USA, November 2007.
5. R. Benmokhtar, B .Huet. K-Space at TRECVid 2008, *Proceedings of TREC*, Gaithersburg-USA, November 2008.

Démonstration

1. T. Athanasiadis, N. Simou, G. Papadopoulos, R. Benmokhtar, K. Chandramouli, V. Tzouvaras, V. Mezaris, M. Phiniketos, Y. Avrithis, I. Kompatsiaris, B. Huet, E. Izquierdo. Combining Segmentation and Classification Techniques for Fuzzy Knowledge-based Semantic Image Annotation, *Proceedings of SAMT*, Koblenz-Germany, December 2008.

Bibliographie

[AB96] B. Achermann and H. Bunke. Combination of classifiers on the decision level for face recognition. *Technical repport of Bern University*, 1996.

[ABB+02] J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, and P. Pala. Soccer highlights detection and recognition using HMMs. *Proceedings of the International Conference on Multimedia and Expo*, pages 825–828, August 2002.

[AD91] H. Almuallim and T-G. Dietterrich. Learning with many irrelevant features. *Ninth National Conference on Artificial Intelligence*, pages 547–552, 1991.

[AD94] H. Almuallim and T-G. Dietterrich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, pages 279–306, 1994.

[Ada07] T. Adamek. Extension of MPEG-7 low-level visual descriptors for TRECVid07. Kspace Technical Report, FP6-027026, 2007.

[AM02] C. Ambroise and G-J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. In *Proceedings of the National Academy of Sciences*, volume 10, pages 6562–6566, 2002.

[App93] A. Appriou. *Formulation et traitement de l'incertain en analyse multi-senseurs*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 1993.

[AQ07] S. Ayache and G. Quénot. TRECVid 2007 collaborative annotation using active learning. *TRECVid, 11th International Workshop on Video Retrieval Evaluation, Gaithersburg, USA*, November 2007.

[ASP+09] T. Athanasiadis, N. Simou, G. Papadopoulos, R. Benmokhtar, K. Chandramouli, V. Tzouvaras, V. Mezaris, M. Phiniketos, Y. Avrithis, Y. Kompatsiaris, B. Huet, and E. Izquierdo. Integrating image segmentation and classification for fuzzy knowledge-based multimedia indexing. In *MMM, 15th International MultiMedia Modeling Conference, Sophia Antipolis-France*, pages 263–274, January 2009.

[ASS00] E-L. Allwein, R-E. Schapire, and Y. Singer. Reducing multiclass to binary : A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1 :113–141, 2000.

[Aya07] S. Ayache. *Indexation de documents vidéos par concepts par fusion de caractéristiques audio, image et texte*. PhD thesis, Institut national polytechnique de Grenoble, France, July 2007.

[Ber04] S-A. Berrani. *Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision : application la recherche d'images par le contenu.* PhD thesis, Université de Rennes 1, France, February 2004.

[BFG+96] J-R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C-F. Shu. The virage image search engine. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, 1996.

[BGH07] R. Benmokhtar, E. Galmar, and B. Huet. Eurecom at TRECVid 2007 : Extraction of high level features. In *TRECVid, 11th International Workshop on Video Retrieval Evaluation, Gaithersburg-USA*, November 2007.

[BGRS99] K-S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "Nearest Neighbor" meaningful ? In *Proceedings of the 7th International Conference on Database Theory*, pages 217–235, 1999.

[BH06] R. Benmokhtar and B. Huet. Classifier fusion : combination methods for semantic indexing in video content. In *ICANN, International Conference on Artificial Neural Networks, Athens-Greece*, pages 65–74, September 2006.

[BH07] R. Benmokhtar and B. Huet. Neural network combining classifier based on Dempster-Shafer theory for semantic indexing in video content. In *MMM, International MultiMedia Modeling Conference, Singapore*, pages 196–205, January 2007.

[BH08a] R. Benmokhtar and B. Huet. Low-level feature fusion models for soccer scene classification. In *IEEE International Conference on Multimedia & Expo, Hannover-Germany*, pages 1329–1332, June 2008.

[BH08b] R. Benmokhtar and B. Huet. Perplexity-based evidential neural network classifier fusion using MPEG-7 low-level visual features. In *MIR, ACM International Conference on Multimedia Information Retrieval, Vancouver-Canada*, pages 336–341, October 2008.

[BH09a] R. Benmokhtar and B. Huet. Hierarchical ontology-based video shots indexing using MPEG-7 visual descriptors. In *CBMI, 7th International Workshop on Content-Based Multimedia Indexing, Crete-Greece*, June 2009.

[BH09b] R. Benmokhtar and B. Huet. Ontological reranking approach for hybrid concept similarity-based video shots indexing. In *WIAMIS, International Workshop on Image Analysis for Multimedia Interactive Services, London-UK*, May 2009.

[BHJ+05] O. Bayer, S. Höhfeld, F. Josbächer, N. Kimm, I. Kradepohl, M. Kwiatkowski, C. Puschmann, M. Sabbagh, N. Werner, and U. Vollmer. Evaluation of an ontology-based knowledge-management-system. a case study of convera retrievalware 8.0. *Inf. Serv. Use*, 25 :181–195, 2005.

[BHK97] P-N. Belhumeur, J-P. Hespanha, and D-J. Kriegman. Eigenfaces vs. fisherfaces : recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :711–720, 1997.

[Bis95] C-M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[BKK02] N. Babaguchi, Y. Kawai, and T. Kitahashi. Event based indexing of broadcasted sports video by intermodalcollaboration. *IEEE Transactions on Multimedia*, pages 68–75, 2002.

[BL97] A-L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97 :245–271, 1997.

[Blo03] I. Bloch. *Fusion d'informations en traitement du signal et des images*. Hermes-Science, Lavoisier, 2003.

[BSM+00] P. Browne, A-F. Smeaton, N. Murphy, S. Marlow, and C. Berrut. Evaluating and combining digital video shot boundary detection algorithms. In *IMVIP - Irish Machine Vision and Image Processing Conference*, 2000.

[CD02] R. Cabasson and A. Divakaran. Automatic extraction of soccer video highlights using a combination of motion and audio features. *In Symposium on Electronic Imaging : Science and Technology : Storage and Retrieval for Media Databases*, pages 272–276, January 2002.

[CHG02] P. Chang, M. Han, and Y. Gong. Extract highlights from baseball game video with hidden Markov models. *Proceedings of the IEEE International Conference on Image Processing*, pages 609–612, September 2002.

[Cie00] L. Cieplinski. Results of core experiment CT4 on dominant color extension, MPEG ISO/IEC 15938-3, ISO/IEC/JTC1/SC29/WG11/M5775, March 2000.

[CM00] K. Chakrabarti and S. Mehrotra. Local dimensionality reduction : A new approach to indexing high dimensional spaces. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 89–100, 2000.

[CM02] A. Cornuéjols and L. Miclet. *Apprentissage Artificiel : Concepts et Méthodes*. Eyrolles, 2002.

[CT00] N. Cristianini and J-S. Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[CTS94] K-H. Chou, L-G. Tu, and I-S. Shyu. Performances analysis of a multiple classifiers system for recognition of totally unconstrained handwritten numerals. *4th International Workshop on Frontiers of Handwritten Recognition*, pages 480–487, 1994.

[Dau] F. Daumas. Méthodes de normalisation des données. *Revue de Statistique Appliquée*, pages 23–38.

[DDL+90] S-C. Deerwester, S-T. Dumais, T-K. Landauer, G-W. Furnas, and R-A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6) :391–407, 1990.

[Den00] T. Denoeux. A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 30(2) :131–150, 2000.

[DH] P. Demartines and J. Herault. Curvilinear component analysis : A self-organizing neural network for nonlinear mapping of data sets. *IEEE transactions on neural networks*, 8.

[DHS01] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2001.

[DK82] P-A. Devijver and K. Kittler. *Pattern recognition : A statistic approach*. Prentice Hall, London, 1982.

[DLY97] M. Dash, H. Liu, and J. Yao. Dimensionality reduction of unsupervised data. In *Ninth IEEE International Conference on Tools with AI*, pages 532–539, 1997.

[DN93] E. Davalo and P. Naim. *Des réseaux de neurones*. Eyrolles, 1993.

[Doh00] D-L. Dohono. High-dimensional data analysis : The curses and blessings of dimensionality. In *The 21st American Mathemathical Society Conference*, 2000.

[DP88] D. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4 :244–264, 1988.

[DT00] R-P. Duin and D-M. Tax. Experiements with classifier combining rules. *Proc. First Int. Workshop MCS 2000*, 1857 :16–29, 2000.

[Eid03] H. Eidenberger. How good are the visual MPEG-7 features ? In *Proceedings SPIE Visual Communications and Image Processing Conference*, volume 5150, pages 476–488. Morgan Kaufmann Publishers, 2003.

[EYGSS99] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C-Y. Suen. An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8) :752–760, 1999.

[Fak90] K. Fakunaga. *Introduction to statistical pattern recognition (2nd edition)*. Academic Press, New York, 1990.

[FBF+94] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4) :231–262, 1994.

[Fel98] C. Felbaum. *WordNet : An electronic lexical database*. Bradford Books, 1998.

[FGL07] J. Fan, Y. Gao, and H. Luo. Hierarchical classification for automatic image annotation. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–118, 2007.

[FH98] P. Felzenszwalb and D. Huttenlocher. Efficiently computing a good segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[FL95] C. Faloutsos and K. Lin. Fastmap : a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the*

ACM International Conference on Management of Data, pages 163–174, New York, NY, USA, 1995.

[FS96] Y. Freund and R-E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[GCA98] M. Girolami, A. Cichocki, and S-I. Amari. A common neural network model for unsupervised exploratory data analysis and independent component analysis. *IEEE Transactions on Neural Networks*, 9(6) :1495–1501, 1998.

[GGLL01] J. Gao, J. Goodman, M. Li, and K-F. Lee. Toward a unified approach to statistical language modeling for chinese. In *ACM Transactions on Asian Language Information Processing*, 2001.

[Gru93] T-R. Gruber. *Towards principles for the design of ontologies used for knowledge sharing in formal ontology in conceptual analysis and knowledge representation.* Kluwer Academic Publishers, 1993.

[GS00] S-A. Glantz and B-K. Slinker. *Primer of applied regression & analysis of variance.* McGraw-Hill/Appleton & Lange, 2000.

[Guy03] I. Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[HCC+05] A-G. Hauptmann, M. Christel, R. Concescu, J. Gao, Q. Jin, W-H. Lin, J-Y. Pan, S-M. Stevens, R. Yan, J. Yang, and Y. Zhang. CMU Informedia's TREC-Vid 2005 Skirmishes. In *TREC Video Retrieval Evaluation Online Proceedings*, 2005.

[Her90] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of Acoustical Society of America*, 87 :1738–1752, 1990.

[HJ94] J. Hèrault and C. Jutten. *Réseaux de neurones et traitement du signal.* Hèrmes, 1994.

[HLB99] A. Hanjalic, R-L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4) :580–588, 1999.

[Ho92] T-K. Ho. *A theory of multiple classifier systems and its application to visual and word recognition.* PhD thesis, State University of New York at Buffalo, 1992.

[HS06] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786) :504 – 507, 2006.

[HSHB05] F. Hönig, G. Stemmer, C. Hacker, and F. Brugnara. Revising perceptual linear prediction (PLP). In *Proceedings of Interspeech'05*, pages 2997–3000, 2005.

[IRB02] K. Idrissi, J. Ricard, and A. Baskurt. An objective performance evaluation tool for color based image retrieval systems. In *International Conference on Image Processing*, pages 389–392, 2002.

[JC97] J. Jiang and D-W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics*, September 1997.

[JDM00] A-K. Jain, R-P. Duin, and J. Mao. Combination of weak classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1), 2000.

[Jol00] J-M. Jolion. *L'indexation*, volume 4. Hermès, 2000.

[Jou03] L. Jourdan. *Méta-heuristiques pour l'extraction de connaissances : Application à la génomique*. PhD thesis, Université USTL de Lille, France, November 2003.

[KBD01] L-I. Kuncheva, J-C. Bezdek, and R-P. Duin. Decision templates for multiple classifier fusion : An experiemental comparaison. *Pattern Recognition*, 34 :299–314, 2001.

[KDLF97] V. Kobla, D-S. Doermann, K-I. Lin, and C. Faloutsos. Compressed domain video indexing techniques using DCT and motion vector information in MPEG video. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, volume 3022, pages 200–211, February 1997.

[KJ97] R. Kohavi and G. John. Wrappers for feature subset selection. *AIJ Journal special issue on relevance*, pages 1–43, 1997.

[KR92] K. Kira and L-A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Conference in Machine Learning*. Morgan Kaufmann Publishers, 1992.

[Kru56] J-B. Kruskal. On the shortest spanning subtree of a graph. pages 48–50, 1956.

[KS06] M. Koskela and A. Smeaton. Clustering-based analysis of semantic concept models for video shots. In *Proceedings of the International Conference on Multimedia and Expo*, pages 45–48, 2006.

[KSL07] M. Koskela, A-F. Smeaton, and J. Laaksonen. Measuring concept similarities in multimedia ontologies : Analysis and evaluations. *IEEE Transactions on Multimedia*, 9(5) :912–922, August 2007.

[KSp] KSpace. Knowledge space of semantic inference for automatic annotation and retrieval of multimedia content. *http ://kspace.qmul.net :8080/kspace/*.

[Kun03] L-I. Kuncheva. Fuzzy versus nonfuzzy in combining classifiers designed by bossting. *IEEE Transactions on fuzzy systems*, 11(6), 2003.

[KY01] E. Kasutani and A. Yamada. The MPEG-7 color layout descriptor : A compact image feature description for high-speed image/ video retrieval. In *International Conference on Image Processing*, volume 1, pages 674–677, 2001.

[LBM03] Y. Li, Z-A. Bandar, and D. Mclean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4) :871–882, 2003.

[LC98] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. *An Electronic Lexical Database*, pages 265–283, 1998.

[Lef07] G. Lefebvre. *Sélection et fusion de signatures visuelles parcimonieuses : Application à la classification d'images naturelles*. PhD thesis, Université Victor Segalen Bordeaux 2, France, December 2007.

[LG03] B. Li and K. Goh. Confidence-based dynamic ensemble for image annotation and semantics discovery. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 195–206, New York, USA, 2003.

[Lin98] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.

[LMJ04] E. Lefevre, J-P. Manata, and D. Jolly. Classification par la théorie de l'évidence pour la gestion de tournée de véhicules. *Congrès Francophone pour le reconnaissance des formes et intelligence artificielle*, 2004.

[LMO04] J. Laaksonen, M. Moskela, and E. Oja. Class distributions on SOM surfaces for feature extraction and object retrieval. *Neural Network*, 17 :1121–1133, 2004.

[LS00] D-D. Lee and H-S. Seung. Algorithms for non-negative matrix factorization. volume 13, pages 556–562, 2000.

[LS02] B. Li and M-I. Sezan. Event detection and summarization in american football broadcast video. *In Symposium on Electronic Imaging : Science and Technology : Storage and Retrieval for Media Databases*, 4676 :202–213, January 2002.

[LS03] C. Liu and H-Y. Shum. Kullback-Leibler boosting. In *International Conference of Computer Vision*, volume 1, pages 587–594, 2003.

[LTS03] C-Y. Lin, B-L. Tseng, and J-R. Smith. Videoannex : IBM MPEG-7 annotation tool for multimedia indexing and concept learning. *Proceedings of the International Conference on Multimedia and Expo*, July 2003.

[LVW02] M. Lazarescu, S. Venkatesh, and G. West. On the automatic indexing of cricket using camera motion parameters. *Proceedings of the International Conference on Multimedia and Expo*, pages 809–812, August 2002.

[LZQ03] T. Liu, H-J. Zhang, and F. Qi. A novel video key-frame extraction algorithm based on perceived motion energy model. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(10) :1006–1013, October 2003.

[MBE01] D-S. Messing, P. Van Beek, and J-H. Errico. The MPEG-7 color structure descriptor : image description using color and local spatial information. In *International Conference on Image Processing*, volume 1, pages 670–673, 2001.

[MK01] A-M. Martinez and A-C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2) :228–233, 2001.

[MM96] B-S. Manjunath and W-Y. Ma. Texture features for browsing and retrieval of image data. 18(8) :837–842, 1996.

[MOVY01] B-S. Manjunath, J. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11(6) :703–715, 2001.

[Mpe01a] Coding of moving pictures and associated audio, MPEG ISO/IEC 15938-3, ISO/IEC/JTC1/SC29/WG11/N4062, March 2001.

[Mpe01b] Information technology - multimedia content description interface-part 3 : Visual, MPEG ISO/IEC 15938-3, ISO/IEC/JTC1/SC29/WG11/N4358, July 2001.

[MSS02] B-S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7 : Multimedia content description interface*. John Wiley and Sons, 2002.

[MWK+05] C. Matuszek, M. Witbrock, Robert C. Kahlert, J. Cabral, D. Schneider, P. Shah, and D. Lenat. Searching for common sense : Populating Cyc from the Web. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, 2005.

[NKFH98] M-R. Naphade, T. Kristjansson, B. Frey, and T-S. Huang. Probabilistic multimedia objects (multijects) : A novel approach to video indexing and retrieval in multimedia systems. In *Proceedings of the IEEE International Conference on Image Processing*, pages 536–540, 1998.

[NKH+06] M-R. Naphade, L. Kennedy, A. Hauptmann, S-F. Chang, and J-R. Smith. LSCOM lexicon definitions and annotations version 1.0. *DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia, 217-2006-3 Columbia University ADVENT Technical Report*, 2006.

[NKK+05] M-R. Naphade, L. Kennedy, J-R. Kender, S-F. Chang, J-R. Smith, P. Over, and A. Hauptmann. A Light Scale Concept Ontology for Multimedia Understanding for TRECVid 2005 (LSCOM-Lite). *IBM Research Technical Report*, November 2005.

[NMB+98] C. Nastar, M. Mitschke, N. Boujemaa, C. Meilhac, H. Bernard, and M. Mautref. Retrieving images by content : The surfimage system. In *MIS : Proceedings of the 4th International Workshop on Advances in Multimedia Information Systems*, pages 110–120, London, UK, 1998. Springer-Verlag.

[NSR01] S. Nepal, U. Srinivasan, and G. Reynolds. Automatic detection of goal segments in basketball videos. *In Proceedings of ACM Multimedia*, pages 261–269, September 2001.

[OIKS05] P. Over, T. Ianeva, W. Kraaij, and A-F. Smeanton. TRECVid 2005 - an overview. In *TRECVid, 11th International Workshop on Video Retrieval Evaluation, Gaithersburg-USA*, November 2005.

[Ope] OpenCV. Intelcorporation : Open source computer vision library : Reference manual, http ://opencvlibrary.sourceforge.net.

[PBH09] M. Paleari, R. Benmokhtar, and B. Huet. Evidence theory based multimodal emotion recognition. In *MMM, 15th International MultiMedia Modeling Conference, Sophia Antipolis-France*, pages 435–446, January 2009.

[PBRT99] J. Puzicha, J-M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *Proceedings of the International Conference on Computer Vision*, volume 2, page 1165. IEEE Computer Society, 1999.

[PJKKK95] P. Paalanen, J. Ilonen J-K. Kamarainen, and H. Kalviainen. Feature representation and discrimination based on Gaussian mixture model probability densities. *Research Report, Lappeenranta University of Technology*, 1995.

[PJW00] D-K. Park, Y-S. Jeon, and C-S. Won. Efficient use of local edge histogram descriptor. In *ACM Workshop on Multimedia*, pages 51–54, 2000.

[PMJDK02] M. Petkovic, V. Mihajlovic, M. Jonker, and S. Djordjevic-Kajan. Multi-modal extraction of highlights from TV formula 1 programs. *Proceedings of the International Conference on Multimedia and Expo*, pages 817–820, August 2002.

[PPS94] A. Pentland, R. Picard, and S. Sclaroff. Photobook : Content-based manipulation of image databases. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, February 1994.

[Pri57] R-C. Prim. Shortest connection networks and some generalizations. In *Bell System Technical Journal*, pages 1389–1401, 1957.

[Que01] G. Quenot. TREC-10 shot boundary detection task : Clips system description and evaluation. In *TRECVid, 5th International Workshop on Video Retrieval Evaluation, Gaithersburg-USA*, 2001.

[Ram01] Michèle Rambaut. Fusion : état de l'art et perspectives. *Convention DSP 99.60.078*, 2001.

[Res95] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[Res99] P. Resnik. Semantic similarity in a taxonomy : An information based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11 :95–130, 1999.

[RHC99] Y. Rui, T. Huang, and S. Chang. Image retrieval : Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4) :39–62, April 1999.

[Rio07] O. Rioul. *Théorie de l'information et du codage*. Hermes-Science, Lavoisier, 2007.

[RKK+01] Y-M. Ro, M. Kim, H-K. Kang, B-S. Manjunath, and J. Kim. MPEG-7 homogeneous texture descriptor. *Electronics and Telecommunications Research Institute*, 23(2) :41–51, 2001.

[RMBB89] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1) :17–30, 1989.

[RR03] S. Raudys and F. Roli. The behavior knowledge space fusion method : Analysis of generalization error and strategies for performance improvement. *Multiple Classifier Systems*, pages 55–64, 2003.

[RS05] M. Rautiainen and T. Seppanen. Comparison of visual features and fusion techniques in automatic detection of concepts from news video. In *Proceedings of the International Conference on Multimedia and Expo*, pages 932–935, 2005.

[Sap90] G. Saporta. *Probabilités analyse des données et statistique*. Technip, 1990.

[SBM+05] E. Spyrou, H. Le Borgne, T. Mailis, E. Cooke, Y. Avrithis, and N-E. O'Connor. Fusing MPEG-7 visual descriptors for image classification. pages 847–852, 2005.

[SBM07] T. Slimani, B. BenYaghlane, and K. Mellouli. Une extension de mesure de similarité entre les concepts d'une ontologie. In *International Conference on Sciences of Electronic, Technologies of Information and Telecommunications*, pages 1–10, March 2007.

[SC99] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–213, 1999.

[SD98] M. Skurichina and R. Duin. Bagging for linear classifiers. In *Proceedings of the Pattern Recognition*, volume 31, pages 909–930, 1998.

[Sha76] G. Shafer. *A Mathematical Theory of Evidence*. 1976.

[SMD02] X. Sun, B-S. Manjunath, and A. Divakaran. Representation of motion activity in hierarchical levels for video indexing and filtering. In *International Conference on Image Processing*, pages 149–152, 2002.

[Sme94] P. Smets. The transferable belief model. *Artif. Intell.*, 66(2) :191–234, 1994.

[Smi] L. Smith. An introduction to neural networks. *Center for Cognitive and Computational Neuroscience, Dept. of Computing & Mathematics, Unversity of Stirling*.

[SO05] D. Sadlier and N. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE transactions on circuits systems and video technology*, pages 1225–1233, Jully 2005.

[Sou05] F. Souvannavong. *Indexation et recherche de plans vidéo par le contenu sémantique*. PhD thesis, Institut Eurécom, France, June 2005.

[SSM98] B. Scholkopf, A. Smola, and K-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem, 1998.

[STKR97] D. Saur, Y-P. Tan, S-R. Kulkarni, and P-J. Ramadge. Automated analysis and annotation of basketball video. *In Symposium on Electronic Imaging : Science and Technology : Storage and Retrieval for Media Databases*, pages 176–187, January 1997.

[SVH04] N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in WordNet. In *Proceedings of European Conference on Artificial Intelligence*, 2004.

[SWA+00] A. Smeulders, M. Worring, S. Antini, A. Gupta, and R. Jain. Content-based image retrieval : The end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1349–1380, 2000.

[Tol06] S. Tollari. *Indexation et recherche d'images par fusion d'informations textuelles et visuelles*. PhD thesis, Université de Sud Toulon-Var, France, October 2006.

[Tor03] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, pages 1415–1438, 2003.

[TRE] TRECVID. Digital video retrieval at NIST. *http ://www-nlpir.nist.gov/projects/trecvid/.*

[TZ04] W. Tavanapong and J. Zhou. Shot clustering techniques for story browsing. *IEEE Transactions on Multimedia*, 6(4) :517–527, August 2004.

[UMY91] H. Ueda, T. Miyatake, and S. Yoshizawa. Impact : An interactive natural-motion-picture dedicated multimedia authoring system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 343–350, New York, NY, USA, 1991.

[Vap00] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 2000.

[VD00] M. Vissac and J-L. Dugelay. Un panorama sur l'indexation d'images fixes. *Journal d'automatique, d'informatique et de traitement du signal*, 2000.

[VF02] M. Vajihollahi and R. Farahbod. The MPEG-7 : Visual standard for content description, School of Computing Science - Simon Fraser University, June 2002.

[VTW04] T. Volkmer, S. Tahaghoghi, and H-E. Williams. RMIT university at TREC 2004. In *TRECVid, 8th International Workshop on Video Retrieval Evaluation, Gaithersburg, USA*, 2004.

[WKSS96] H. Wactlar, T. Kanade, M-A. Smith, and S-M. Stevens. Intelligent access to digital video : The informedia project. *IEEE Computer*, 29(5), 1996.

[WMS00] P. Wu, B-S. Manjunath, and H-D. Shin. Dimensionality reduction for image retrieval. volume 3, pages 726–729, 2000.

[WP94] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994.

[WSB98] R. Weber, H-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 194–205, 1998.

[WTS04] Y. Wu, B-L. Tseng, and J-R. Smith. Ontology-based multi-classification learning for video concept detection. *Proceedings of the International Conference on Multimedia and Expo*, 2 :1003–1006, June 2004.

[XF07] L. Xi and K. Fukui. Fisher non-negative matrix factorization with pairwise weighting. In *Machine Vision Applications*, volume 1, pages 380–383, 2007.

[XJK01] E-P. Xing, M-I. Jordan, and R-M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608. Morgan Kaufmann, 2001.

[XKS92] L. Xu, A. Krzyzak, and C-Y. Suen. Methods of combining multiple classifiers and their application to hardwriting recognition. *IEEE Trans.Sys.Man.Cyber*, 22 :418–435, 1992.

[XZ06] F. Xu and Y-J. Zhang. Evaluation and comparison of texture descriptors proposed in MPEG-7. *Journal of Visual Communication and Image Representation*, 17 :701–716, 2006.

[YLK+01] W. You, K-W. Lee, J-G. Kim, J. Kim, and O-S. Kwon. Content-based video retrieval by indexing object's motion trajectory. *IEEE International Conference on Consumer Electronics*, pages 352–353, 2001.

[YM98] D. Yining and B-S. Manjunath. Netra-V : Toward an object-based video representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5) :616–627, September 1998.

[YY96] M. Yeung and B-L. Yeo. Time-constrained clustering for segmentation of video into story unites. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 375–380, 1996.

[YYHM98] Z. Yueting, R. Yong, T-S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 866–870, 1998.

[YYYL95] D. Yow, B. Yeo, M. Yeung, and G. Liu. Analysis and presentation of soccer highlights from digital video. *In Proceedings of the Second Asian Conference on Computer Vision*, pages 499–503, December 1995.

[ZD95] L-M. Zouhal and T. Denoeux. An adaptive k-NN rule based on Dempster-Shafer theory. In *Proc. of the 6th Int. Conf. on Computer Analysis of Images and Patterns*, pages 310–317. Springer Verlag, 1995.

[ZKS93] H-J. Zhang, A. Kankanhalli, and S-W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1) :10–28, June 1993.

[ZL03] D. Zhang and G. Lu. Evaluation of MPEG-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9(1) :15–30, 2003.

[ZMM99] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, 7(2) :119–128, March 1999.