République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb de Blida

Faculté des Sciences

Département d'informatique

Mémoire de fin d'étude

Pour l'Obtention du diplôme de Master en Informatique

Option : Ingénierie de Logiciel

RÉSUMÉ VIDÉO MULTI-SOURCES

Réalisé par :

Nom : Benteftifa

Prénom : Kheireddine

Nom : Bersali

Prénom : Mahmoud

Mr. Président : Benyahia Mohamed

Mr. Examinateur : Nehal Djillali

Mr. Promoteur : Kameche Abdallah Hicham

Soutenu le : Lundi 2 Juillet 2018

Remerciement

Avant tout, nous tenons à remercier Dieu tout puissant de nous avoir donné la force, la patience, la foi, et le courage pour accomplir notre travail.

Nous tenons à exprimer nos profondes gratitudes, sincères remerciements et toute notre reconnaissance à l'égard de notre promoteur Monsieur Kameche Abdallah Hichem pour avoir accepté de diriger ce travail et pour ses précieux conseils, sa patience, et surtout sa générosité.

Nous exprimons notre grande reconnaissance aux professeurs de département d'informatique de l'université de Blida 1 qui nous ont tant donné pour être ce que nous sommes aujourd'hui.

Par ailleurs, nous tenons à remercier vivement les membres de jury qui ont fait l'honneur d'accepter de participer en tant qu'examinateurs à notre soutenance. Pour finir, merci a à toute personne ayant aidé de près ou de loin à la réalisation de ce travail.

Résumé

La vidéosurveillance est un système de surveillance par des caméras qui peuvent être installées dans les espaces publics afin de gérer les risques. Pour utiliser efficacement ces caméras, l'opérateur doit regarder les images et répondre à des activités suspectes. Des opérateurs humains entrainés et expérimentés peuvent faire efficacement ce suivi, mais seulement pour un nombre limité de vidéos. Vu la croissance rapide d'énorme flux de vidéos qui se trouvent sur les ordinateurs nécessite le développement de nombreux outils pour leur manipulation tel que le *«résumé vidéo»*.

La plupart des travaux actuels se focalisent généralement sur la construction du résumé d'une seule vidéo, seuls quelques-uns se sont portés au problème de résumés multi-vidéos où la prise en compte d'autres contraintes et éléments s'impose, nous citons par exemple le fait que plusieurs informations sont présentes d'une façon similaire dans diverses vidéos.

Dans ce mémoire, nous proposons une solution qui consiste à développer une application pour la génération de résumé vidéo multi-sources basé sur l'apprentissage profond pour l'extraction des vecteurs caractéristiques profondes et l'utilisation d'une architecture neuronale basée sur les réseaux de neurones récurrents à longue « mémoire court-terme » (LSTM) qui prend les fonctionnalités spatio-temporelles présentes dans les images de la vidéo pour la génération dynamique du résumé final.

Mots clés : *Résumé vidéo, apprentissage profond, réseau de neurones récurrents*

Abstract

Video surveillance is a surveillance system by using cameras that can be installed in public spaces in order to manage risk. To effectively use these cameras, the operator must view the images and respond to suspicious activities. Trained and experienced human operators can do this effectively, but only for a limited number of videos. Given the rapid growth of huge video streams on computers, many tools for manipulation such as video summary are needed.

Most of the current work focuses on the construction of the summary of a single video, only a few have addressed the problem of multi-video summaries where the consideration of other constraints and elements is necessary, we quote for example the fact that several pieces of information are present in a similar way in various videos.

In this thesis, we propose a solution that consists in developing an application for the generation of multi-video summarization based on deep learning for the extraction of deep features vectors and the use of a neural architecture based on recurrent neural networks with long "short-term memory" (LSTM) that takes the spatial-temporal functionalities present in the video images for the dynamic generation of the final summary.

Keywords : *video summarization, deep learning, recurrent neural networks*

ملخص

المراقبة بالفيديو هو نظام مراقبة يعتمد على كاميرات موضوعة في مساحات عمومية لإدارة المخاطر.

للاستخدام الفعال لهذه الكاميرات، يجب على المستخدم أن ينظر الى الصور والاستجابة للنشاط المشبوه، يمكن للمدربين وذوي الخبرة أن يتتبعوا ذلك بفعالية، ولكن فقط لعدد محدود من مقاطع الفيديو

نظرا لتراكم السريع للفيديوهات المتواجدة على أجهزة الكمبيوتر مما يستوجب تطوير طرق للتحكم بها مثل ملخص الفيديو

معظم الأبحاث تعطي الاهتمام عادة بخلاصة فديو واحد، قليل من اهتم بمشكل خلاصة متعددة للفيديوهات وتأخذ بعين الاعتبار القيود والعناصر المطلوبة، نذكر على سبيل المثال عدة معلومات متكررة في مختلف أشرطة الفيديو

في هذا العمل، نقترح تطوير تطبيق لخلاصة الفيديو متعدد المصدر يعتمد على التعليم المعمق من أجل استخراج أشعة الخصائص المعمقة واستخدام بنية عصبية مبنية على الشبكات العصبية المتكررة مع "ذاكرة قصيرة طويلة المدى" والتي تأخذ الوظائف المكانية والزمانية الموجودة في صور الفيديو للتوليد الديناميكي لخلاصة الفيديو النهائي

الكلمات المفتاحية : ملخص فيديو ، التعليم المعمق ، شبكة عصبية متكررة

Table des matières

Page

Liste des tableaux 7

Table des figures 8

1 Résumé Vidéo : Concepts de base liés à la vidéo 3

1.1 Introduction 4

1.2 Résumé statique et résumé dynamique 4

1.3 Structure des vidéo 5

1.3.1 Le plan 6

1.3.2 Coupe (Shot) 7

1.4 Le signal vidéo : de l'analogique au numérique 8

1.4.1 Signal analogique 8

1.4.2 Signal numérique 9

1.5 Nombre d'images par seconde et résolution 9

1.6 Extraction des caractéristiques 10

1.7 Descripteurs des caractéristiques 11

1.7.1 Détecteurs des points d'intérêt 11

1.7.2 Descripteurs SIFT (Scale Invariant Feature Transformation) 11

1.7.3 Descripteur SURF (Speeded Up Robust feature) 12

1.7.4 Le descripteur Histogramme de Gradient Orienté (HoG) 13

1.8 Conclusion 14

2 Apprentissage automatique, un domaine de l'intelligence artificielle 15

2.1 Introduction 16

2.2 Les types d'apprentissage automatique 16
2.2.1 Apprentissage supervisé 17
2.2.2 Apprentissage non-supervisé 18
2.2.3 Apprentissage semi-supervisé 18
2.2.4 Apprentissage par renforcement 18
2.3 Quelques algorithmes d'apprentissage supervisé : 19
2.3.1 Arbre de décision 19
2.3.2 Les machines à vecteurs de support 20
2.3.3 La méthode des K plus proches voisins (K nearest neighbor) 21
2.4 Les réseaux de neurones 22
2.4.1 Rétropropagation de l'erreur (*backpropagation*) 23
2.4.2 Réseau de neurones profonds 24
2.4.3 Réseaux de neurones à convolution (CNN ou ConvNet) 25
2.4.4 Réseaux de neurones récurrents (RNN) 30
2.4.5 Réseaux antagonistes génératifs (generative adversarial networks ou GANs) 33
2.4.6 Les Autoencodeurs 35
2.5 Conclusion 37

3 TRAVAUX PROPOSÉ POUR LA GÉNÉRATION DES RÉSUMÉS VIDÉOS 38
3.1 Introduction 39
3.2 Travaux basés sur une seule source vidéo 39
3.2.1 Méthodes basées sur l'échantillonnage 39
3.2.2 Méthodes basées sur les plans 40
3.2.3 Méthodes basées sur les scènes (ou macro-segments) 41
3.2.4 Approches basées sur l'extraction d'événements intéressants 42
3.3 Travaux basé sur plusieurs sources vidéos 42
3.3.1 Approche basé sur la théorie des graphes : 43
3.3.2 Approches basées sur les caractéristiques spatio-temporelles 44
3.4 Conclusion 45

4 Approche proposé **46**

4.1 Introduction 47

4.2 Vue globale de l'approche 48

4.3 Phase de pré-traitement 49

4.4 Phase d'extraction de caractéristiques 50

4.4.1 Le modèle C3D (Conv3D) 51

4.4.2 Architecture du modèle C3D 52

4.5 Phase de réduction de dimension 53

4.6 Création du résumé 54

4.7 Conclusion 57

5 Présentation des résultats **58**

5.1 Introduction 59

5.2 Environnement matériel 59

5.3 Environnement logiciel 59

5.3.1 Python 59

5.3.2 Numpy 60

5.3.3 OpenCV 60

5.3.4 TensorFlow 61

5.3.5 Keras 61

5.3.6 Le format HDF5 61

5.4 Ensemble de donnée (Dataset) 61

5.4.1 Office 62

5.4.2 Lobby 62

5.4.3 Campus 63

5.5 Outil de mesure 63

5.6 Résultat et discussion : 64

5.7 Conclusion 65

Bibliographie **67**

LISTE DES TABLEAUX

TABLE **Page**

5.1 Calcul des paramètres Recall, Precision est F-mesure 63

5.2 Comparaison de performance par rapport aux différentes approches(Précision, Rappel, F-measure) 64

TABLE DES FIGURES

FIGURE		Page
1.1	Représentation des deux méthodes de résumé vidéo	5
1.2	Structure d'une vidéo	6
1.3	Chaque ligne illustre un exemple d'images de même scène	7
1.4	Coupe par fondu enchaîné	7
1.5	Coupe par changement progressif de l'éclairage	8
1.6	coupe par changement brusque du plan	8
1.7	Représentation d'un signal analogique [Media, 2000]	8
1.8	Représentation d'un signal numérique	9
1.9	représentation d'un signal numérique type binaire	9
1.10	Types de points clés, (De gauche à droite) marche, toit, coin, ligne ou bord, arête ou contour, région maxima	11
1.11	Descripteur SIFT	12
1.12	(a) Image originale, (b) Après application du descripteur SIFT	12
1.13	(a) Image originale, (b) Après application du descripteur SURF	13
1.14	(a) Image originale, (b) Après application du descripteur HoG	13
2.1	Exemple d'abre de décision	20
2.2	Exemple d'un hyperplan séparateur [Mohamadally Hasan, 2006]	21
2.3	Exemple de vecteurs de support [Mohamadally Hasan, 2006]	21
2.4	Exemple de fonctionnement de la méthode des k-plus proches voisins	22
2.5	Schéma d'un neurone biologique	23
2.6	Perceptron multi-couches (MLP)	24
2.7	Illustration simplifié d'un réseau de neurone profond	25

2.8 ConvNet pour la reconnaissance des objets 25
2.9 Un réseau de neurone à convolution (ConvNet) 26
2.10 L'opération de convolution 27
2.11 Application de la fonction d'activation ReLU 28
2.12 Max Pooling 28
2.13 Exemple de réseau de neurone récurrent 30
2.14 Visualisation synthétique de la propagation de l'information lors de la passe avant dans une couche LSTM 32
2.15 Réseau récurrent à portes (GRU) 33
2.16 Generative Adversarial Network (GAN) 34
2.17 à gauche : des images réelle (à partir d'Imagenet), à droite : images générés par le réseau génératif 34
2.18 Illustration simplifié d'un AutoEncoder 35
2.19 Exemple d'un DAE sur des chiffres manuscrits 36
2.20 Reconstruction des chiffres manuscrites avec les VAE 36
2.21 Reconstruction d'images par VAE prises à partir d'un ensemble de test séparé . . . 37

3.1 Un aperçu du résumé vidéo multi-vue d'après 43
3.2 Représentation d'événements similaires généré par les random walk 44
3.3 Quelques événements résumés sur le dataset Office Lobby produit par 44

4.1 Illustration d'un réseau de caméra multi vues 47
4.2 Une illustration d'un réseau de caméras multi-sources 48
4.3 Schéma globale de notre approche 49
4.4 Schéma globale de la phase prétraitement 50
4.5 Schéma globale de la phase d'extraction des caractéristiques profondes 51
4.6 Différence entre l'opération de convolution 2D et convolution 3D 51
4.7 Architecture du ConvNet3D 52
4.8 Déconvolution des caractéristiques apprises par le C3D sur la couche conv2a . . . 53
4.9 Déconvolution des caractéristiques apprises par le C3D sur la couche conv3b . . . 53
4.10 Représentation d'un simple autoencodeur 54

4.11 Processus de décision d'un agent avec l'apprentissage par renforcement 55

5.1 Une image d'exemple du dataset Office 62

5.2 Une image d'exemple du dataset Lobby 63

Introduction générale

Les architectures de deep learning (apprentissage profond) modernes ont atteint des performances compétitives sur de nombreuses tâches relatives. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatique du langage naturel.

Parmi l'une des tâches de vision par ordinateur, on retrouve le *résumé vidéo*.

Beaucoup de tâches de surveillance visuelle, telle que le résumé vidéo, sont accomplies en analysant les caractéristiques des images. Cependant, cette analyse se révèle être insuffisante, voire parfois incorrectes dans le cadre de la vidéosurveillance publique. En effet, les données récoltées des caméras de vidéosurveillance publique différent selon l'angle de vue, de l'heure et des conditions environnementales, et souvent, les événements « intéressants » passent inaperçus.

Afin d'avoir une meilleure sémantique pour un résumé vidéo , il serait donc intéressant de combiner les informations collectées depuis plusieurs angles de vue et prendre en considération des données non visuelles, telles que les rapports météorologiques, la vitesse de circulation et le nombres d'événements pour complémenter l'analyse et le résumé vidéo.

L'objectif principal est de concevoir et d'implémenter une solution de résumé vidéo multi-sources basée sur la notion d'apprentissage profond, une technique prometteuse dans ce contexte afin de générer un résumé d'une haute qualité.

Pour cela cette mémoire est organisé comme suit :

- **Chapitre I :** Une description de la structure des différents composants et des caractéristiques d'une vidéo sont présentés suivi d'une étude de quelques descripteurs d'image.
- **Chapitre II :** Dans ce chapitre, nous nous intéresserons à présenter les différents types d'ap-

prentissage automatique de façon plus particulière les réseaux de neurones.

- **Chapitre III :** : Ce chapitre contient une étude comparative des travaux proposés dans la littérature pour la génération des résumés vidéo.
- **Chapitre IV :** Dans ce chapitre nous allons décrire la méthode que nous avons proposée pour la génération du résumé vidéo à partir de plusieurs vidéos.
- **Chapitre V :** Ce chapitre contient la présentation des résultats obtenus.

CHAPITRE 1

RÉSUMÉ VIDÉO : CONCEPTS DE BASE LIÉS À LA VIDÉO

Sommaire

1.1 Introduction 4
1.2 Résumé statique et résumé dynamique 4
1.3 Structure des vidéo 5
1.3.1 Le plan 6
1.3.2 Coupe (Shot) 7
1.4 Le signal vidéo : de l'analogique au numérique 8
1.4.1 Signal analogique 8
1.4.2 Signal numérique 9
1.5 Nombre d'images par seconde et résolution 9
1.6 Extraction des caractéristiques 10
1.7 Descripteurs des caractéristiques 11
1.7.1 Détecteurs des points d'intérêt 11
1.7.2 Descripteurs SIFT (Scale Invariant Feature Transformation) 11
1.7.3 Descripteur SURF (Speeded Up Robust feature) 12
1.7.4 Le descripteur Histogramme de Gradient Orienté (HoG) . . 13
1.8 Conclusion 14

1.1 Introduction

Avec l'augmentation continue du volume de données audiovisuelles, la recherche d'information et de documents pertinents devient un véritable défi. [Guironnet, 2006] Ces énormes quantités de contenu audiovisuel ont largement dépassé la capacité que possède l'être humain de les visualiser et ont rendu difficile la recherche de contenus intéressants pour un utilisateur. Le besoin d'outils efficaces permettant aux utilisateurs de choisir le contenu qu'ils vont regarder est donc manifeste. [Boukadida, 2015] L'utilisateur a besoin d'avoir un aperçu des vidéos qu'il souhaite regarder. Cet aperçu sera utile pour gagner un temps considérable et avoir une idée claire sur le contenu vidéo. Le résumé vidéo permet de répondre à ce besoin en fournissant une version courte de la vidéo qui doit contenir l'essentiel de l'information, tout en étant le plus concis possible.

Dans ce chapitre nous présentons une description de la structure des vidéos, après la mise en évidence de l'importance et l'utilité des résumés vidéos, nous citons les principales méthodes que constituent les *deux* grandes familles de résumé de vidéo : issu d'une sélection d'images représentatives (*images clés*). Résumé dynamique, résultant d'une sélection de segments extraits de la vidéo, équivalent à une bande annonce. [Guironnet, 2006]

1.2 Résumé statique et résumé dynamique

Avec le développement des technologies récentes, la sauvegarde de documents audio et vidéo augmente, mais il devient de plus en plus difficile de trouver le temps pour revoir ces documents. Ces quantités énormes de données multimédia ont de loin dépassé la capacité que nous avons à toutes les traiter, et en tirer avantage. Cette situation s'accentue au fil du temps, et de nouveaux outils plus « *intelligents* » pour faire face à ce problème deviennent indispensables. Les résumés vidéos permettent d'avoir rapidement une idée sur le contenu de très grandes bases de vidéos, sans nécessiter la visualisation et l'interprétation de l'ensemble des vidéos. [Yahiaoui, 2003]

Deux sortes de résumé peuvent être retrouvées : (comme montré dans la figure 1.1)

- **Résumé statique :** qui est construit sous forme d'un ensemble d'images représentatives qui

sont soigneusement extraites à partir de la vidéo, Ces images sont appelées **images-clés** (*key frame*). Cette représentation peut nous permettre d'avoir un accès direct aux différentes parties du document vidéo original. Elle peut également faire partie d'une interface interactive de recherche de données par le contenu. Les images composant le résumé seront alors considérées comme des index, ou des images requêtes pour les utilisateurs de cette interface. Cette collection d'images procure ainsi en un clin d'œil une idée générale et globale des éléments pertinents compris dans cette vidéo. Cependant, cette représentation ne permet pas de capturer le dynamisme et la continuité des images d'une séquence vidéo. [Yahiaoui, 2003]

- **Résumé dynamique :** qui consiste à construire une séquence visuelle d'une durée désirée qui permet de préserver l'information temporelle des segments extraits de la vidéo. Ce type de résumé dynamique représente une version réduite du flux visuel de la vidéo entière. L'utilisateur est obligé de regarder la totalité du résumé d'une durée prédéfinie pour avoir une idée générale et comprendre le contenu de la vidéo originale. [Yahiaoui, 2003]

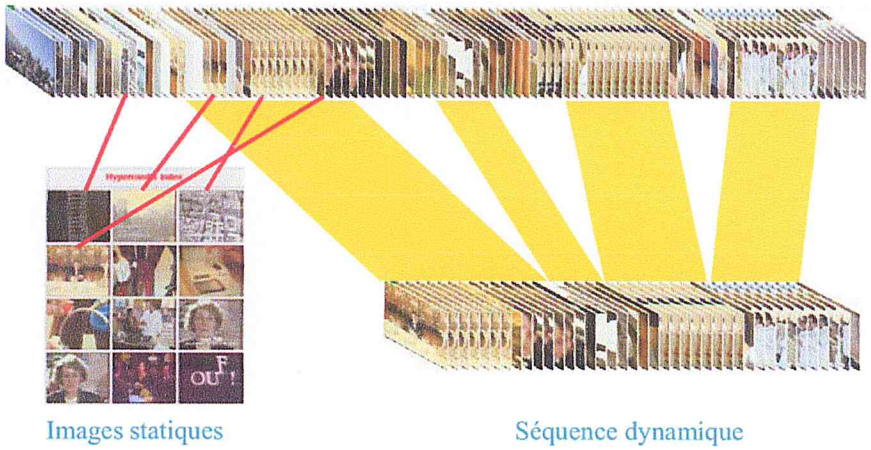


FIGURE 1.1 – Représentation des deux méthodes de résumé vidéo [Yahiaoui, 2003]

1.3 Structure des vidéo

Une vidéo se compose d'une succession d'images affichées à une certaine fréquence (25 ou 30 images) par seconde, généralement accompagnées d'une bande son c'est à dire de données audio synchronisées.

Certains auteurs [Xiong *et al.*, 2006] distinguent *deux* types de vidéo :

— Les vidéos ayant un script (scripted content), il s'agit de vidéos qui possèdent une structure bien définie comme les journaux télévisés et les films.

— Les vidéos sans script (unscripted content), soit aucun script n'est écrit comme dans les vidéos de sport ou la surveillance vidéo.

1.3.1 Le plan

Souvent considèré comme l'unité de base des vidéos, se définit comme une portion de vidéo filmée continument sans effets spéciaux ni coupure. A partir du plan, différents niveaux de segmentation (voir figure 1.2) ont été proposés : [Guironnet, 2006]

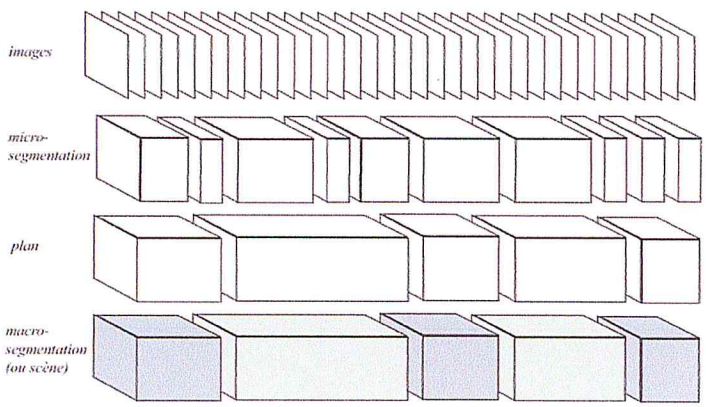


FIGURE 1.2 – Structure d'une vidéo [Guironnet, 2006]

1. Les images adjacentes à l'intérieur de chaque plan sont regroupées suivant une caractéristique commune (par exemple, si elles ont un même mouvement de caméra) pour former une micro-segmentation, premier niveau de la segmentation.
2. Le dernier niveau de la segmentation consiste à réunir des micro-segments pouvant provenir de plans différents pour établir une macro-segmentation ou scène (voir figure 1.3). Une **scène** correspond à un regroupement de plans se déroulant dans un même lieu. [Guironnet, 2006] (voir figure 1.3)

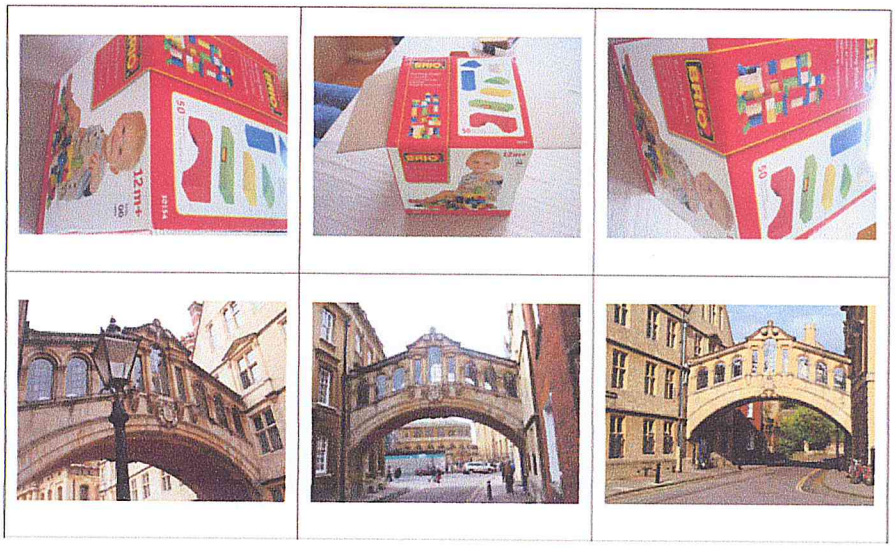


FIGURE 1.3 – Chaque ligne illustre un exemple d'images de même scène

1.3.2 Coupe (Shot)

Elle est définie comme étant une transition immédiate d'une scène à l'autre qui se produit entre deux plans [Porter *et al.*, 2001]. Par définition, une transition correspond au point de jonction entre deux plans. Il existe plusieurs types de transitions dans les vidéos. Celles-ci ont été regroupées suivant *deux* grandes familles de transitions :

1. **Les changements progressifs :** qui consistent en l'obtention d'une continuité visuelle lors du passage d'un plan à l'autre. Cette transition est réalisée :

   — soit par fondu enchaîné [Guironnet, 2006]

FIGURE 1.4 – Coupe par fondu enchaîné [Guironnet, 2006]

   — soit par changement progressif de l'éclairage jusqu'à atteindre une teinte uniforme (*fade in/fade out*)

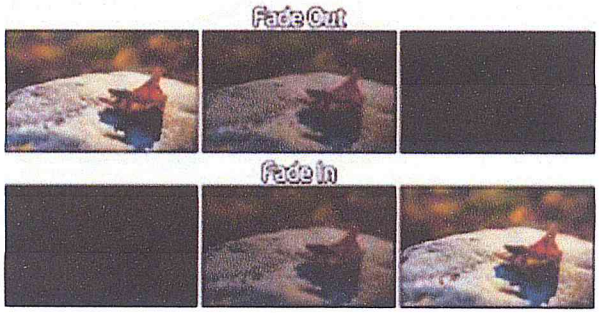


FIGURE 1.5 – Coupe par changement progressif de l'éclairage [Mostefaui Souad, 2015]

2. **Les changements de plans brusques (ou instantanée) :** qui consistent à juxtaposer la fin d'un plan avec le début du plan suivant sans transition [Porter *et al.*, 2001].

FIGURE 1.6 – coupe par changement brusque du plan [Mostefaui Souad, 2015]

1.4 Le signal vidéo : de l'analogique au numérique

L'un des premiers concepts que nous devons assimiler est la distinction entre vidéo analogique et vidéo numérique pour cela nous allons définir c'est quoi un signal analogique et un signal numérique :

1.4.1 Signal analogique

La télévision (support d'affichage vidéo le plus communément répandu) fonctionne en mode analogique. Les images vidéo affichées lui sont transmises sous forme de signal analogique, par l'intermédiaire des ondes ou du câble. Les signaux analogiques sont constitués d'ondes qui changent constamment. Autrement dit, le signal, à un instant donné, peut prendre n'importe quelle valeur comprise entre le minimum et le maximum autorisés. [Media, 2000]

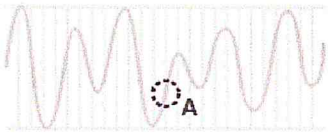


FIGURE 1.7 – Représentation d'un signal analogique [Media, 2000]

1.4.2 Signal numérique

Les signaux numériques en revanche, sont exclusivement transmis sous forme de points sélectionnés par intervalles sur la courbe *(voir figure 1.8)*

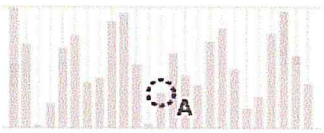


FIGURE 1.8 – Représentation d'un signal numérique [Media, 2000]

Un ordinateur peut utiliser un signal numérique de type binaire, qui décrit ces points sous la forme d'une suite de valeurs minimales ou maximales correspondant respectivement au « zéro » et au « un ».

Cette suite de zéros et de uns peut ensuite être interprétée à la réception comme un ensemble de nombres représentatifs de l'information émise à l'origine. [Media, 2000]

FIGURE 1.9 – représentation d'un signal numérique type binaire [Media, 2000]

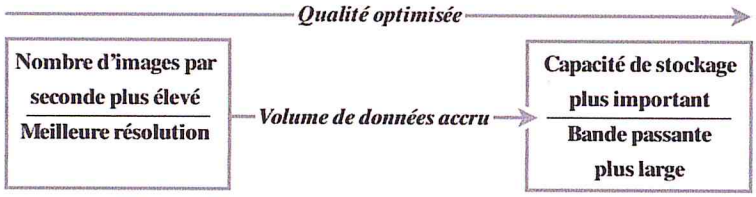
1.5 Nombre d'images par seconde et résolution

Lorsque l'œil humain perçoit une suite d'images séquentielles, il se produit un phénomène étonnant. Si les images sont affichées suffisamment rapidement, l'œil ne distingue pas chacune d'entre elles séparément, mais perçoit une légère animation. C'est sur cette base que sont élaborés les films et les vidéos. La cadence de l'animation est désignée sous le terme de nombre d'images par seconde. Pour qu'une légère animation, soit perceptible à l'œil, une cadence d'environ 10 images par seconde est nécessaire. [Media, 2000]

Cependant, la qualité des vidéos ne dépend pas seulement du nombre d'images par seconde. La quantité d'informations contenues dans chaque image est également déterminante. Elle est désignée sous le terme de résolution d'image. La résolution correspond en règle générale au nombre d'éléments individuels constituant l'image (pixels) affichés à l'écran. Elle

est exprimée sous la forme du nombre de pixels utilisés sur l'axe horizontal de l'image multiplié par le nombre de pixels utilisés sur l'axe vertical (par exemple, 640 x 480 ou 720 x 480). Toutes choses étant égales par ailleurs, une résolution plus élevée permet d'obtenir une image de meilleure qualité.

Le nombre d'images par seconde et la résolution sont des paramètres très importants en matière de vidéo numérique, car ils déterminent le volume de données à transmettre et à enregistrer en vue de la diffusion.



1.6 Extraction des caractéristiques

Un problème important dans la conception des résumés est le choix des caractéristiques qui représentent le contenu des images, ceux-ci sont souvent des descripteurs de bas niveau. [Guironnet, 2006] : (principalement en termes de couleurs, textures et formes). Il y a principalement deux approches pour les caractéristiques qui peuvent être extraites. La première est la construction de descripteurs globaux à toute l'image. Dans ce cas, il s'agit de fournir des observations sur la totalité de l'image. L'avantage des descripteurs globaux est la simplicité des algorithmes mis en œuvre, et le nombre réduit d'observations que l'on obtient. Cependant, l'inconvénient majeur de ces descripteurs est la perte de l'information de localisation des éléments de l'image. La seconde approche est locale consiste à calculer des attributs sur des portions restreintes de l'image. L'avantage des descripteurs locaux est de conserver une information localisée dans l'image, évitant ainsi que certains détails ne soient noyés par le reste de l'image. L'inconvénient majeur de cette technique est que la quantité d'observations produite est très grande, ce qui implique un gros volume de données à traiter. [Saïda, 2011]

1.7 Descripteurs des caractéristiques

De nombreux algorithmes de vision par ordinateur reposent sur la localisation de points d'intérêt ou de points clés dans chaque image et le calcul d'une description d'entité à partir de la région de pixels entourant le point d'intérêt.

1.7.1 Détecteurs des points d'intérêt

Le point d'intérêt est le point d'ancrage et fournit souvent les attributs d'invariance d'échelle, de rotation et d'illumination pour le descripteur; Le descripteur ajoute plus de détails et d'autres attributs d'invariance. Les groupes de points d'intérêt et les descripteurs décrivent ensemble les objets réels. [Nikita Kaushik, 2016]

Les points-clés peuvent être considérés comme un ensemble composé de coins, arêtes ou contours, et de plus grandes caractéristiques ou régions telles que les blobs comme la montre la figure suivante :

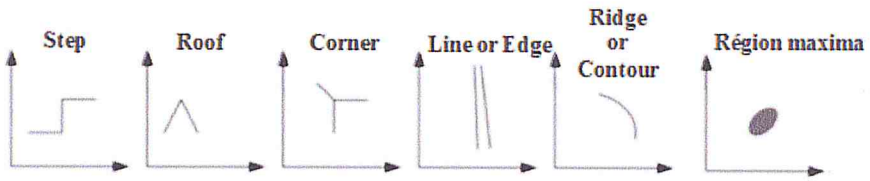


FIGURE 1.10 – Types de points clés, (De gauche à droite) marche, toit, coin, ligne ou bord, arête ou contour, région maxima [Nikita Kaushik, 2016]

Les algorithmes utilisés pour trouver les points d'intérêt peuvent être appelés détecteurs, et les algorithmes utilisés pour décrire les caractéristiques peuvent être appelés descripteurs.

Il existe plusieurs descripteurs comme SIFT (transformation de l'entité invariante à l'échelle), SURF (fonctionnalités robustes accélérées) et HOG (histogramme du dégradé).

1.7.2 Descripteurs SIFT (Scale Invariant Feature Transformation)

Qui se traduit par la transformation de caractéristiques visuelles invariante à l'échelle ,l'idée général de cette méthode est de transformer une image en vecteurs de caractéristiques, lesquels doivent être dans l'idéal invariants aux transformations géométriques (rotation, mise

à l'échelle), et dans une moindre mesure invariants à l'illumination. Il s'agit de détecter des points remarquables (ou clés) (voir figure 1.11), qui vont permettre d'identifier un objet. Le point fort de la méthode de Lowe est qu'elle est capable de mettre en correspondance des points distants avec des variations de caméra importantes. [Lowe, 2004]

L'algorithme des SIFT se divise en 2 grandes étapes :

— Calcul des points d'intérêts et des descripteurs

— Mise en correspondance « *matching* »

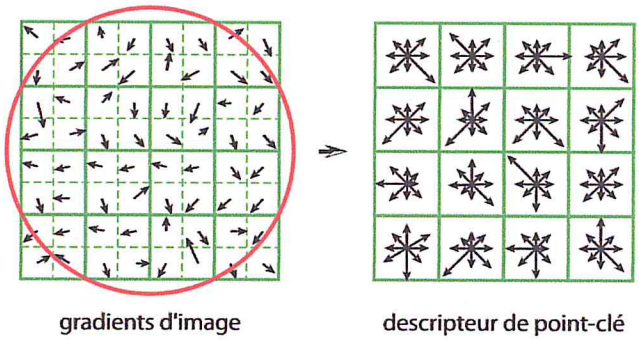


FIGURE 1.11 – Descripteur SIFT [Nikita Kaushik, 2016]

La première étape permet de transformer une image en vecteurs de caractéristiques. La seconde étape consiste en la comparaison des descripteurs de deux images afin de détecter un objet en particulier ou de trouver quelle transformation à subi une image.

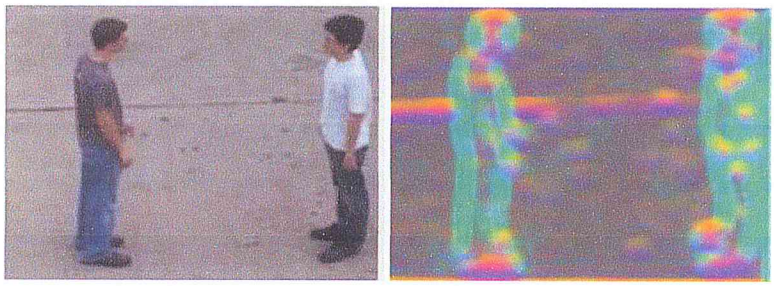
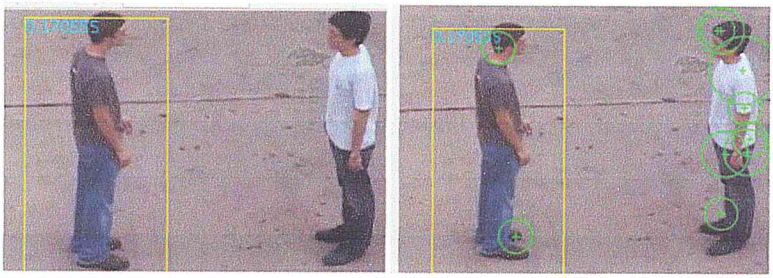


FIGURE 1.12 – (a) Image originale, (b) Après application du descripteur SIFT [Nikita Kaushik, 2016]

1.7.3 Descripteur SURF (Speeded Up Robust feature)

Constituent une bonne alternative aux SIFT. Cette méthode s'appuie largement sur les SIFT, mais qu'il le surpasse en rapidité, et se révèle plus robuste quant a certaines transforma-

tions d'images. SURF a la particularité d'être moins couteux que SIFT en utilisant notamment une méthode d'évaluation des gradients basée sur les ondelettes de Haar (voir figure 1.13). Il est également utilisé pour la reconnaissance d'objets, l'enregistrement, la reconstruction 3D et la classification. [Bay *et al.*, 2008]

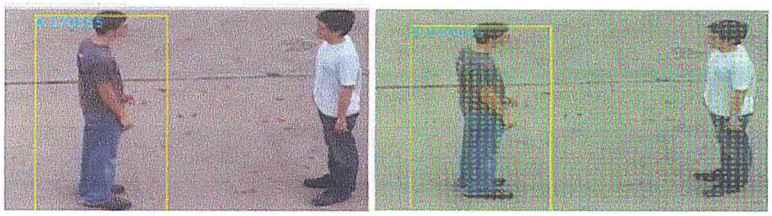


(a) (b)

FIGURE 1.13 – (a) Image originale, (b) Après application du descripteur SURF [Nikita Kaushik, 2016]

1.7.4 Le descripteur Histogramme de Gradient Orienté (HoG)

Ce descripteur est utilisé dans le traitement des images et la vision par ordinateur pour la détection des objets. L'idée essentielle derrière l'histogramme de gradient orienté c'est que l'apparence locale et la forme d'objet dans une image peut être décrite par la distribution d'intensité des gradients ou de direction des contours. La mise en œuvre de ce descripteur peut être obtenue en divisant l'image en petites régions connectées, appelées cellules (voir figure 1.14) , et pour chaque cellule on calcule un histogramme des directions de gradient ou des orientations de contour pour les pixels dans la cellule. La combinaison de ces histogrammes représente alors le descripteur.



(a) (b)

FIGURE 1.14 – (a) Image originale, (b) Après application du descripteur HoG [Nikita Kaushik, 2016]

Le descripteur HoG maintient quelques avantages clés par rapport aux autres méthodes.

Puisque le descripteur histogramme de gradient orienté opère sur les cellules localisées, la méthode maintient l'invariance à des transformations géométriques et photométriques, ces changements ne feront leur apparition que dans les larges régions d'espaces. [Dalal and Triggs, 2005]

1.8 Conclusion

Nous nous sommes intéressés dans ce chapitre aux caractéristiques des vidéos. Nous avons présenté, dans un premier temps, la notion de résumé vidéo et ses deux méthodes qui sont les résumé **statiques** à base de sélection d'images clés appelé *keyframe*, ainsi que les résumé **dynamique** à base d'une sélection d'extrait de la vidéo, puis nous avons abordé la structure de base des vidéos, suivi d'une brève étude sur les différents descripteurs des caractéristiques, qui seront représentées sous formes de vecteurs que nous allons les détailler dans les chapitres suivants.

CHAPITRE 2

APPRENTISSAGE AUTOMATIQUE, UN DOMAINE DE L'INTELLIGENCE ARTIFICIELLE

Sommaire

2.1 Introduction 16

2.2 Les types d'apprentissage automatique 16

2.3 Quelques algorithmes d'apprentissage supervisé : 19

2.4 Les réseaux de neurones 22

2.5 Conclusion 37

2.1 Introduction

L'intelligence artificielle (I.A.) est une discipline scientifique qui étudie la façon de créer des programmes dits intelligents. Par programmes intelligents, on veut généralement parler de programmes capables de résoudre des problèmes traditionnellement considérés comme étant propres aux capacités humaines.

L'apprentissage automatique (*machine learning*) cherche à permettre à l'ordinateur d'imiter la capacité humaine d'apprendre à partir d'exemples, lui donnant la possibilité d'agir sans être explicitement programmé. En général, ce domaine se concentre sur les algorithmes qui apprennent à partir d'exemples pour ensuite permettre de généraliser sur de nouveaux exemples non observés auparavant. L'apprentissage automatique est maintenant une composante importante de plusieurs domaines tels que le traitement automatique du langage naturel, la reconnaissance d'objets, la reconnaissance de la parole, la bio-informatique et bien d'autres encore. [Lauly, 2016]

L'apprentissage automatique se divise en plusieurs types se distinguant par la nature des tâches devant être apprises : l'apprentissage supervisé, l'apprentissage non-supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement. Nous allons les détailler dans ce qui suit.

Ce chapitre s'intéresse donc à l'utilité de l'apprentissage automatique pour la génération du résumé vidéo. Nous allons dans un premier temps présenter les différents types d'apprentissage automatique suivi de quelques modèles (algorithmes) d'apprentissage automatique et enfin une étude sur les réseaux de neurones.

2.2 Les types d'apprentissage automatique

Il existe plusieurs types différents d'apprentissage automatique, qui se distinguent essentiellement par leur objectif, *i.e.*, la nature de ce qui doit être appris. Bien qu'ils puissent trouver application dans des contextes différents, ces types d'apprentissage peuvent aussi être combinés dans un même système, on va les détailler dans ce qui suit .

2.2.1 Apprentissage supervisé

Généralement le but pour un modèle en apprentissage automatique est de générer une fonction de prédiction $f(x)$ à partir d'un ensemble de données D fourni, qu'on appelle ensemble d'entraînement. Dans le cadre de l'apprentissage supervisé, D est composé de paires d'exemples (x, y), où x est un vecteur servant d'entrée au modèle et y un vecteur cible qui représente ce que l'on veut prédire. Le fait d'avoir une cible pour chaque exemple, et de s'en servir, est la caractéristique principale de l'apprentissage supervisé. [Lauly, 2016] Supposons que la fonction $f(x)$, prenant en entrée le vecteur x de taille J, ait la forme suivante :

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_J x_J \tag{2.1}$$

Où θ_0 à θ_J sont les paramètres du modèle d'apprentissage. Le but de ce modèle est donc de trouver les valeurs des paramètres afin d'obtenir les meilleures prédictions possible, *c'est-à-dire* le plus proche possible des cibles.

Dans beaucoup de cas, les données **étiquetées** (données avec cibles) sont générées par des personnes qui assignent manuellement une cible à chaque exemple. Cependant, le fait de créer des données étiquetées ne nécessite pas toujours une intervention humaine. Quelle que soit la façon dont les données ont été obtenues, on les passe ensuite à un algorithme d'apprentissage qui cherche à modéliser la relation entre les entrées et leurs cibles. Il existe une autre catégorisation importante en apprentissage automatique, liée à la nature même des cibles, qui est le type de problèmes à résoudre. Nous allons présenter les deux types les plus utilisés dans la littérature. Il s'agit des cibles composées de valeurs *discrètes* ou *continues*. [Lauly, 2016]

- **Problème de classification :** Habituellement, pour un problème de classification, l'ensemble de données utilisé possède un nombre fini de classes et chaque exemple est associé à l'une d'elles. La cible de chaque exemple aura une valeur discrète représentant une classe en particulier. Avec de telles données, un modèle en apprentissage automatique apprendra à assigner les classes aux entrées.
- **Problème de régression :** Dans le cadre d'un problème de régression, la cible est composée d'un ou de plusieurs éléments de valeurs continues. Un modèle en apprentissage automatique apprendra à prédire une ou des valeurs réelles.

En météorologie, la prédiction de la température est un bon exemple de problème de régression. En effet, la valeur à prédire ici est continue. On peut aussi ajouter d'autres éléments à la cible comme la pression atmosphérique et le taux d'humidité, créant ainsi un vecteur de valeurs continues. [Lauly, 2016]

2.2.2 Apprentissage non-supervisé

Contrairement à l'approche supervisée, l'ensemble de données D utilisé en apprentissage non-supervisé n'est pas composé de paires d'exemples (x, y), mais seulement de x, *c'est-à-dire* qu'il n'y a plus de cible associée à chaque exemple. Dans un tel cadre, un modèle en apprentissage automatique modélisera l'information fournie en entrée seulement.

Dans l'approche non-supervisée, la fonction $f(x)$ retournée par un algorithme d'apprentissage n'est pas dictée par la nature des données. En effet, contrairement à l'approche supervisée, il n'y a pas de but explicitement exprimé par le biais de cibles.

Le problème devant être résolu par la fonction est donc défini par l'usager. Cependant, peu importe le problème choisi, un modèle non supervisé apprendra toujours des caractéristiques en lien avec la distribution de probabilité générant les données d'entraînement. [Lauly, 2016]

2.2.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé est en fait un mélange des deux approches que l'on vient de présenter, soit l'apprentissage supervisé et non-supervisé. L'apprentissage semi-supervisé concerne le cas où le jeu de données est partiellement étiqueté. L'objectif est d'entrainer un modèle qui soit capable de tirer parti à la fois des cibles présentes mais aussi des données non étiquetés [Thong, 2015].

2.2.4 Apprentissage par renforcement

L'apprentissage par renforcement a comme objectif d'entraîner un agent à se comporter de façon intelligente dans un environnement donné. Un agent interagit avec l'environnement en choisissant, à chaque temps donné, d'exécuter une action parmi un ensemble d'actions permises. Le comportement intelligent que doit apprendre cet agent est donné implicitement via

un signal de renforcement qui, après chaque décision de l'agent, indique s'il a bien ou mal agi. L'agent doit donc se baser sur ce signal afin d'améliorer son comportement, qui est dicté par sa politique d'actions. À chaque temps donné, l'agent a normalement à sa disposition un ensemble de caractéristiques ou indicateurs d'entrée, décrivant l'environnement. [Larochelle, 2009]

Les concepts d'action et de signal de renforcement sont probablement ceux qui distinguent le plus l'apprentissage par renforcement des apprentissages supervisé et non-supervisé. Contrairement à l'apprentissage supervisé, le comportement intelligent à apprendre n'est pas explicitement donné par une cible à prédire mais doit plutôt être défini par l'utilisateur à l'aide d'un signal de renforcement. [Larochelle, 2009]

2.3 Quelques algorithmes d'apprentissage supervisé :

2.3.1 Arbre de décision

La popularité de la méthode arbre de décision repose en grande partie sur sa simplicité. Il s'agit de trouver un partitionnement des individus que l'on représente sous la forme d'un arbre de décision. L'objectif est de produire des groupes d'individus les plus homogènes possibles du point de vue de la variable à prédire. Il est d'usage de représenter la distribution empirique de l'attribut à prédire sur chaque sommet (nœud) de l'arbre. (voir figure 2.1)

L'arbre de décision est composé d'un point d'entrée constitué par le premier test, appelé racine de l'arbre, ce dernier dirige les individus vers différentes branches selon son résultat, branches qui se décomposent à leurs tours grâce à d autres tests, chaque point de connexion entre plusieurs branches est un nœud intermédiaire, pour aboutir aux nœuds terminaux qui sont appelés *feuilles*; les feuilles contiennent les valeurs de la classe. [Souad, 2014]

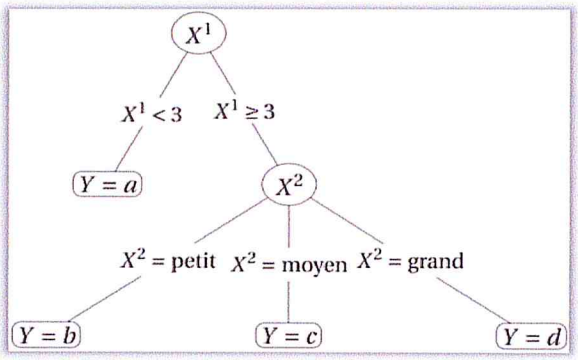


FIGURE 2.1 – Exemple d'abre de décision [Souad, 2014]

Une fois le modèle construit, des règles de classement sont extraites et sont par la suite utilisées pour classer de nouveaux objets pour lesquels la classe est inconnue. L'induction avec des arbres de décision est l'une des formes d'algorithmes d'apprentissages les plus simples et pourtant les plus efficaces. [Souad, 2014]

2.3.2 Les machines à vecteurs de support

L'algorithme des machines à vecteurs de support a été développé dans les années 90 par le russe Vladimir Vapnik. Initialement, les SVM ont été développé comme un algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones) et il a fourni de bons résultats sur des problèmes réels [Mahé, 2003].

L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant d'augmenter la séparabilité des données. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial [Mahé, 2003].

Pour deux classes d'exemples donnés, le but des SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points [Mohamadally Hasan, 2006].

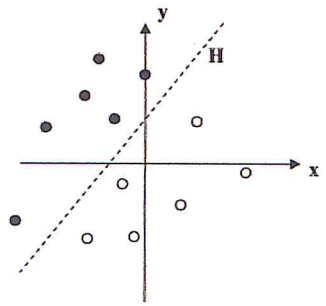


FIGURE 2.2 – Exemple d'un hyperplan séparateur [Mohamadally Hasan, 2006]

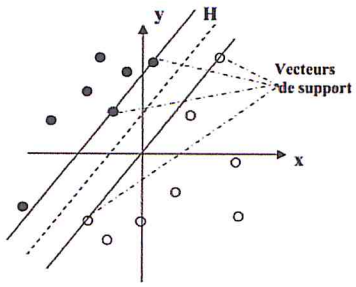


FIGURE 2.3 – Exemple de vecteurs de support [Mohamadally Hasan, 2006]

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support. [Mohamadally Hasan, 2006]

Il existe une infinité d'hyperplans qui peuvent servir de séparateurs. L'idée des machines à vecteurs de support est de choisir le meilleur hyperplan, *c'est-à-dire* celui qui donnera la règle qui se généralisera le mieux à d'autres données que celles de l'ensemble d'apprentissage. [P. Mahé, 2003]

Aujourd'hui, les SVMs sont utilisées dans différents domaines de recherche et d'ingénierie tel que le diagnostic médical, le marketing, la biologie, la reconnaissance de caractères manuscrits et de visages humains. [Abdelhamid, 2012]

2.3.3 La méthode des K plus proches voisins (K nearest neighbor)

La méthode de k plus proches voisins est une méthode de l'apprentissage supervisé de type apprentissage à base d'instances (*instance-based learning*). Son principe est pour une nouvelle instance I de prédire les k objets dans la base d'apprentissage qui sont les plus semblables à I (voir figure 2.4). Cela met en jeu la notion de distance ou similarité entre deux objets. Cet algorithme utilise les objets trouvés pour classer l'instance I en lui attribuant la classe la plus probable. [Hawarah, 2008]

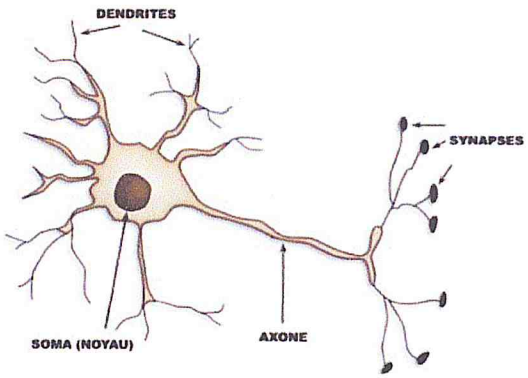


FIGURE 2.5 – Schéma d'un neurone biologique [Glorot, 2014]

Traditionnellement, on attribue la naissance des réseaux de neurones aux travaux de [McCulloch and Pitts, 1943]. Ils ont montré que les réseaux de neurones peuvent réaliser des fonctions logiques et arithmétiques. La structure employée était fondée sur des neurones logiques ou binaires interconnectés, qui ne connaissent que les réponses 0 ou 1. Ce modèle du neurone inspira la création du perceptron de [Rosenblatt, 1958]

Le **Perceptron** de [Rosenblatt] fut le premier réseau apprenant, il fut donc une étape importante dans l'histoire des réseaux de neurones. À partir d'observations étiquetées (contexte d'apprentissage supervisé), il est capable d'ajuster les poids d'un neurone, afin de converger vers une configuration apte à réaliser des opérations de classification ou de généralisation. [Feuilloy, 2009]

2.4.1 Rétropropagation de l'erreur (*backpropagation*)

L'autre moment clé fut la découverte de nouveaux modèles capables de dépasser les limites du perceptron, le plus célèbre est le **perceptron multicouches** (ou MLP pour Multi-Layer Perceptron) (figure 2.6) qui peut être vu comme un ensemble d'unités de traitement, appelés textit-noeuds ou *neurones*, reliées entre elles par des connections pondérées. Les poids de ces connections étant les paramètres du modèle. Ces neurones et ces connections sont organisés en couches : *(i)* La première couche est appelée couche d'entrée, *(ii)* la dernière est appelée couche de sortie et *(iii)* la ou les couches du milieu sont appelées couches cachée. [Baccouche, 2013]

Plus précisément, la découverte ne fut pas le modèle, mais l'algorithme d'apprentissage, qui permit l'utilisation de modèles plus complexes. En effet, *F. Rosenblatt* avait déjà pris conscience de la nécessité de couches cachées dans un réseau, afin d'élargir les capacités du perceptron.

Ainsi, [Rumelhart *et al.*, 1986] ont publié un nouvel algorithme d'apprentissage, appelé l'algorithme de *rétropropagation de l'erreur (backpropagation)*, qui permit l'apprentissage et donc l'optimisation des paramètres de réseaux de neurones à plusieurs couches. [Feuilloy, 2009]

En effet, cette technique nous permet de calculer de façon efficace les gradients de tous les poids d'un réseau de neurones. La rétropropagation porte bien son nom, car le flux de calculs fait le chemin inverse de la propagation avant. Elle commence donc par la sortie et se dirige vers l'entrée. L'idée est de calculer la dérivée de la fonction objectif par rapport à la couche de sortie pour ensuite propager cette information à travers le réseau jusqu'à l'entrée du modèle. Le gradient de chaque couche cachée est exprimé en réutilisant les dérivées des couches qui les suivent. [Lauly, 2016]

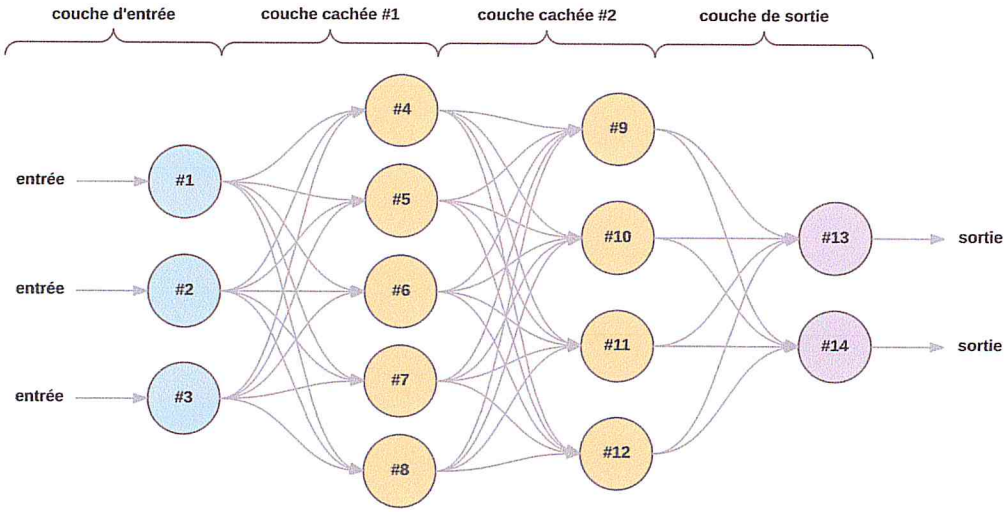


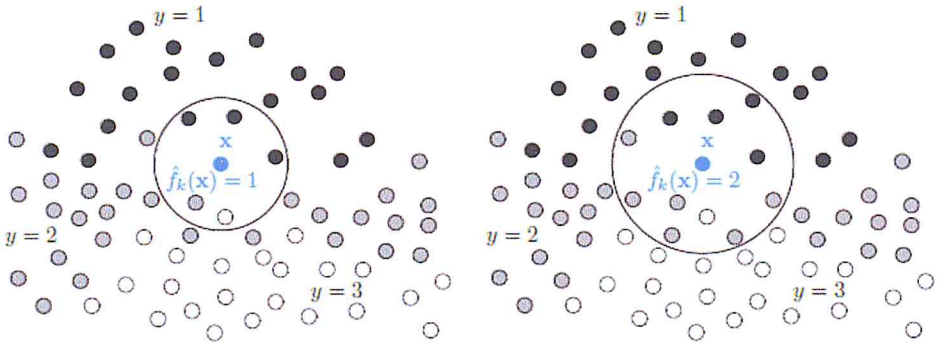
FIGURE 2.6 – Perceptron multi-couches (MLP) [Gelly, 2017]

2.4.2 Réseau de neurones profonds

Les réseaux de neurones dits *"profonds"* (Deep Neural Networks en anglais) sont des MLPs avec un nombre de couches supérieur à trois (comme le montre la figure 2.7). Pendant longtemps les méthodes d'apprentissage pour ce type de réseaux de neurones acycliques ne permettaient pas de converger vers un réseau de neurones performant. Des avancées majeures sur les méthodes d'entraînement et le choix de la fonction de transfert Rectified Linear Unit (ReLU) (détaillé dans le paragraphe 2.4.3.2), qui minimise l'impact de la dilution du gradient dans les couches basses du réseaux ont permis d'utiliser des réseaux de neurones de plus en plus gros. [Gelly, 2017]

On peut résumer l'algorithme comme suit :

```
Début
   On cherche à classer l'instance I
      pour chaque objet J de l'ensemble d'apprentissage faire
         calculer la distance D(J,I) entre J et I
      fin pour
   Dans les k objets les plus proches de I
     calculer le nombre d'occurrences de chaque classe
   Attribuer à I la classe la plus probable
fin
```



(a) k=5 (b) k=11

FIGURE 2.4 – Exemple de fonctionnement de la méthode des k-plus proches voisins pour des valeurs du paramètres $k = 5$ et $k = 11$. On considère trois classes, représentées respectivement en noir ($y = 1$), en gris ($y = 2$) et en blanc ($y = 3$). [Mohammed, 2014]

2.4 Les réseaux de neurones

Les réseaux de neurones sont des assemblages fortement connectés d'unités de calcul, les neurones formels. Ils ont pour origine un modèle du neurone biologique (voir figure 2.5) et ont vocation à imiter certains mécanismes du cerveau humain

Un neurone formel est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie". [Mouelhi-Chibani, 2009]

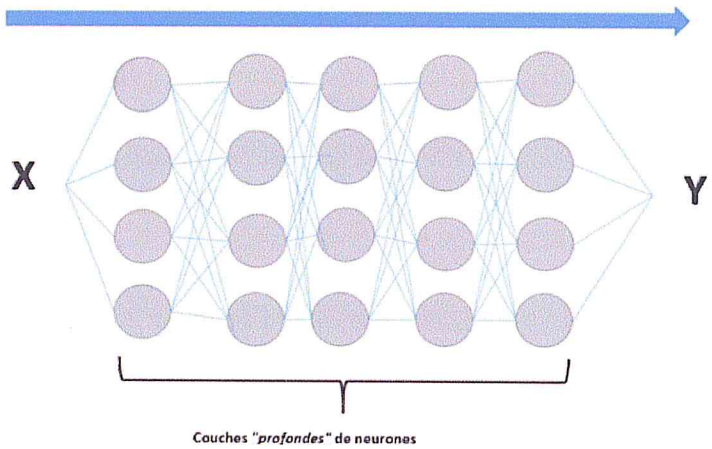


FIGURE 2.7 – Illustration simplifié d'un réseau de neurone profond

2.4.3 Réseaux de neurones à convolution (CNN ou ConvNet)

Les réseaux de neurones à convolution opèrent généralement sur des images et sont l'un des types de réseaux de neurones le plus répandu, il s'agit d'une forme particulière de MLP dont l'architecture des connexions est inspirée de celle du cortex visuel chez les mammifères. [Baccouche, 2013] Ces réseaux sont capables de catégoriser les informations des plus simples aux plus complexes (figure 2.8). Ils consistent en un empilage multicouche de neurones, des fonctions mathématiques à plusieurs paramètres ajustables, qui prétraitent de petites quantités d'informations.

FIGURE 2.8 – ConvNet pour la reconnaissance des objets [Ren *et al.*, 2015]

C'est dans les années 90 que ces réseaux seront popularisés avec les travaux de *Yann. Le Cun* et al sur la reconnaissance de caractères [LeCun *et al.*, 1990]. Les auteurs ont proposé une série de réseaux ConvNets, baptisés LeNet (de 1 à 5) dont l'architecture est similaire à la figure 2.9.

Les ConvNets se sont révélés très efficaces dans des domaines tels que la reconnaissance et la classification de l'image, l'identification des visages, des objets et des panneaux de signalisa-

tion en dehors de l'alimentation de la vision dans les robots et les voitures auto-conductrices [Ren *et al.*, 2015].

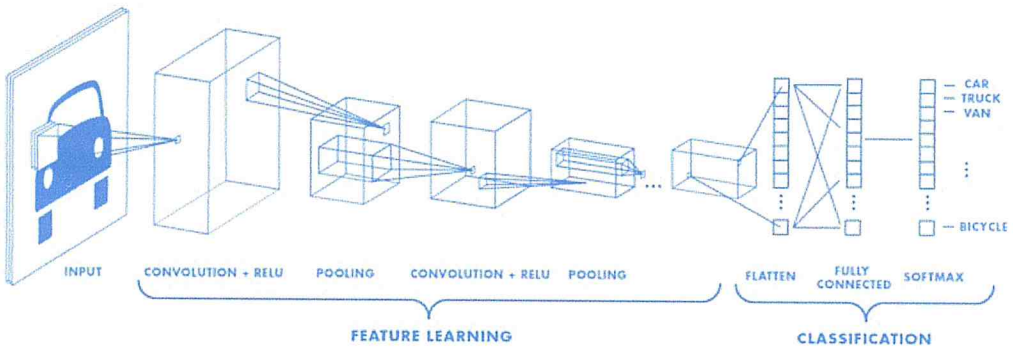


FIGURE 2.9 – Un réseau de neurone à convolution (ConvNet) [Wong, 2017]

Comme le montre la figure ci-dessus, un ConvNet est capable de classer une image d'entrée en plusieurs catégories et d'attribuer correctement la plus grande probabilité selon le contenu de l'image. La somme de toutes les probabilités dans la couche de sortie devrait être égale à 1.

Il existe *quatre* opérations principales dans le ConvNet, illustrées à la figure 2.9

1. Convolution
2. Unités Rectifié Linéaire (ReLU)
3. Pooling ou Sub Sampling
4. Classification (couche entièrement connectée)

Ces opérations sont les éléments constitutifs de base de chaque réseau neuronal convolutif.

2.4.3.1 Étape de convolution

Quand on lui présente une nouvelle image, le CNN ne sait pas exactement si les caractéristiques seront présentes dans l'image ou où elles pourraient être, il cherche donc à les trouver dans toute l'image et dans n'importe quelle position.

En calculant dans toute l'image si une caractéristique est présente, nous faisons un filtrage. Les mathématiques que nous utilisons pour réaliser cette opération sont appelés une convolution (voir figure 2.10) , de laquelle les réseaux de neurones à convolution tiennent leur nom. Les mathématiques derrière le principe de convolution ne sont pas bien complexes. Pour calculer

la correspondance entre une caractéristique et une sous-partie de l'image, il suffit de multiplier chaque pixel de la caractéristique par la valeur que ce même pixel contient dans l'image. Ensuite, on additionne les réponses et divise le résultat par le nombre total de pixels de la caractéristique. Si les 2 pixels sont blancs (de valeur 1) alors $1 * 1 = 1$. Si les deux sont noires, alors $(-1) * (-1) = 1$. Dans tous les cas, chaques pixels correspondant ont pour résultat 1. De manière similaire, chaque décalages donnent -1. [Crouspeyre, 2017]

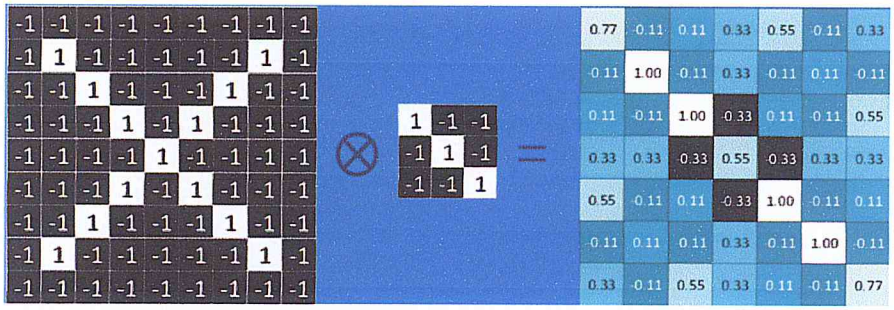


FIGURE 2.10 – L'opération de convolution [Crouspeyre, 2017]

Si tous les pixels dans une caractéristique correspondent, alors leur addition puis leur division par le nombre total de pixels donne 1. De la même manière, si aucun des pixels de la caractéristique ne correspond à la sous-partie de l'image, alors la réponse est -1. Pour compléter une convolution, on répète ce processus alignant les caractéristiques avec chaque sous-partie de l'image. On prend alors le résultat de chaque convolution et créé avec un nouveau tableau de 2 Dimensions, basé sur où dans l'image le patch se trouve. Cette map des correspondances est aussi une version filtrée de l'image d'origine. C'est une map indiquant où la caractéristique a été trouvée dans l'image. [Crouspeyre, 2017]

Dans la terminologie CNN, la matrice 3 × 3 est appelée *«filtre»* ou *«noyau»* ou *«détecteur de caractéristiques»* et la matrice formée en glissant le filtre sur l'image et en calculant le produit en points est appelée « *Convolved Feature* » ou « *Activation Map* » ou « *Feature Map* » . Il est important de noter que les filtres servent de détecteurs de caractéristiques à partir de l'image d'entrée d'origine. [Ren *et al.*, 2015]

2.4.3.2 Unités Rectifié Linéaire (ReLU)

Une fois la convolution effectuée, une fonction d'activation est appliquée sur toutes les valeurs de l'image filtrée. Le choix de la fonction d'activation (*tanh* ou *sigmoid*) peut dépendre du

problème de classification mais la correction Relu est préférable, car il en résulte la formation de réseau neuronal plusieurs fois plus rapide [Krizhevsky *et al.*, 2012].

La fonction *ReLU* est défini comme :

$$f(\sigma) = max(0, \sigma)$$

Le résultat d'une couche *ReLU* est de la même taille que ce qui lui est passé en entrée, avec simplement toutes les valeurs négatives éliminées.

Il s'agit d'un outil fondamental car sans lequel le *CNN* ne produirait pas vraiment les résultats qu'on lui connaît .

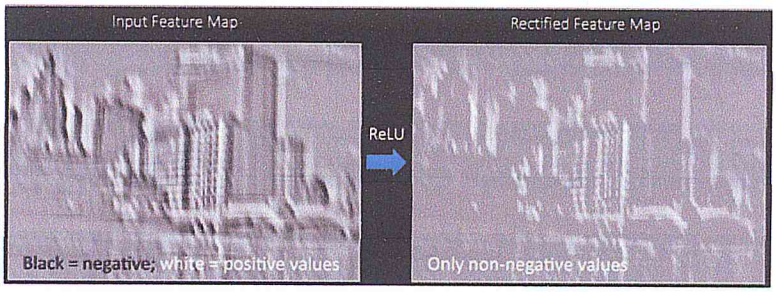


FIGURE 2.11 – Application de la fonction d'activation ReLU [Nefy, 2017]

2.4.3.3 Pooling (ou Sub Sampling)

Le Pooling est une méthode permettant de prendre une large image et d'en réduire la taille tout en préservant les informations les plus importantes qu'elle contient. Les mathématiques derrière la notion de pooling ne sont une nouvelle fois pas très complexe. En effet, il suffit de faire glisser une petite fenêtre pas à pas sur toutes les parties de l'image et de prendre (dans le cas de Max Pooling) la valeur maximum de cette fenêtre à chaque pas (figure 2.12). Au lieu de prendre le plus grand élément, nous pouvons également prendre la moyenne (Pooling moyen) ou la somme de tous les éléments. [Crouspeyre, 2017]

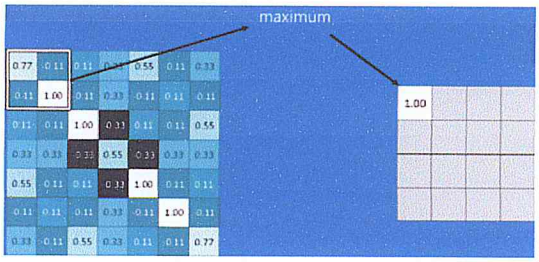


FIGURE 2.12 – Max Pooling [Crouspeyre, 2017]

En pratique, on utilise souvent une fenêtre de 2 ou 3 pixels de côté et une valeur de 2 pixels pour ce qui est de la valeur d'un pas. Après avoir procédé au pooling, l'image n'a plus qu'un quart du nombre de ses pixels de départ.

Parce qu'il garde à chaque pas la valeur maximale contenue dans la fenêtre, il préserve les meilleures caractéristiques de cette fenêtre. Cela signifie qu'il ne se préoccupe pas vraiment d'où a été extraite la caractéristique dans la fenêtre.

Le résultat est que le CNN peut trouver si une caractéristique est dans une image, sans se soucier de l'endroit où elle se trouve. Au final, une couche de pooling est simplement un traitement de pooling sur une image ou une collection d'images. L'output aura le même nombre d'images mais chaque images aura un nombre inférieur de pixels. Cela permettra ainsi de diminuer la charge de calculs. [Crouspeyre, 2017]

2.4.3.4 Couche entièrement connecté (Fully Connected Layer)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux de neurones traditionnels). Ils prennent les images filtrées de haut niveau et les traduisent en votes.

Lorsqu'une nouvelle image est présentée au CNN, elle se répand à travers les couches inférieures jusqu'à atteindre la couche finale entièrement connectée. L'élection a ensuite lieu. Et la solution avec le plus de vote est déclarée comme catégorie de l'image. (voir figure 2.9)

En pratique, plusieurs couches entièrement connectées sont souvent ajoutées les unes à la suite des autres, avec chaque couche intermédiaire votant pour des catégories "dites cachées". En effet, chaque couche additionnelle laisse le réseau apprendre des combinaisons chaque fois plus complexes de caractéristiques qui aident à améliorer la prise de décision. [Crouspeyre, 2017]

- **La fonction softmax :** enfin, on connecte la dernière couche fully connected à une fonction qui se nomme softmax (fonction exponentielle normalisée), cette fonction est utilisée pour produire une distribution de probabilité entre les différentes classes (chaque classe

aura une valeur réelle comprise dans l'intervalle [0, 1]). [Pibre *et al.*, 2016]

2.4.4 Réseaux de neurones récurrents (RNN)

Nous avons présenté lors du paragraphe 2.4.1 les Perceptrons multi-couches, qui représentent une catégorie de modèles neuronaux dits *"acycliques"* (en anglais *feedforward neural network*), c'est à dire dans lesquels les flux d'information ne se propagent que dans un sens : De l'entrée du réseau vers sa sortie. Ces réseaux n'ont donc que des connexions directes, qui ne forment pas de boucles. Si cette contrainte est relâchée, nous obtenons les réseaux de neurones récurrents (RNN pour Recurrent Neural Networks). [Baccouche, 2013]

Autrement ce sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. En cela, ils sont plus proches du vrai fonctionnement du système nerveux, qui n'est pas à sens unique. Ces réseaux possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire : ils peuvent prendre en compte à un instant t un certain nombre d'états passés.

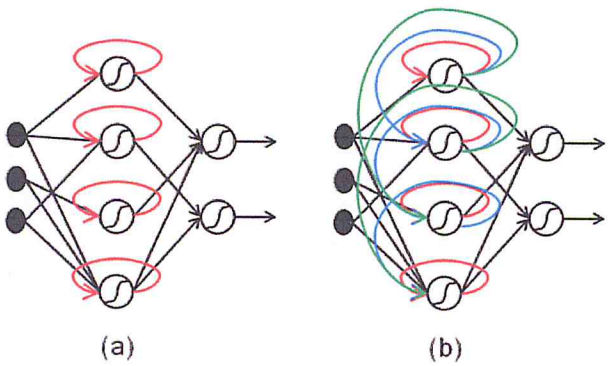


FIGURE 2.13 – Réseau de neurones récurrent avec une couche cachée **(a)** - Réseau auto récurrent basique **(b)** - Réseau récurrent plus complexe. [Baccouche, 2013]

La figure ci dessus illustre le réseau de neurones récurrent : **(a)** le plus basique, dont la couche cachée est dite auto-récurrente (*c'est à dire* que chaque neurone de la couche cachée possède une seule connexion récurrente reliant sa sortie à son entrée). et **(b)** présente quant à elle un exemple de réseau de neurones récurrent plus complexe, où tous les neurones de la couche cachée sont connectés entre eux.

2.4.4.1 Réseaux récurrents à longue mémoire à court terme (LSTM)

L'intérêt majeur des RNNs est leur capacité à utiliser l'information contextuelle lors de l'apprentissage. Toutefois, même si en théorie cette propriété les rend particulièrement adaptés au traitement des séquences, en pratique les RNNs "classiques" sont incapables de traiter des séquences faisant intervenir des écarts temporels supérieurs à 10 instants entre les entrées et les sorties désirées correspondantes. Afin d'y remédier, la solution la plus utilisée dans l'état de l'art est celle des réseaux récurrents à longue mémoire à court terme (LSTM pour long short-term memory) (voir figure 2.14) , consiste en un ensemble de sous-réseaux récurrents particuliers (appelés "blocs de mémoire") situés au niveau de la couche cachée, et contenant chacun une ou plusieurs "cellules de mémoire". [Baccouche, 2013] Cette architecture est définie par deux idées clés que nous allons présenter dans ce qui suit :

- La première idée clé architecturale des réseaux LSTM est l'introduction d'un noeud spécial appelé CEC (pour Constant Error Carousel) qui peut être vu comme une unité qui permet de "collecter" et de "conserver" les informations jugées pertinentes tout au long de la séquence, et de les "présenter" au reste du réseau.
- La seconde idée clé est l'utilisation de "portes" multiplicatives qui sont des fonctions d'activation qui permettant d'ouvrir ou de fermer une connexion donnée. Dans la première version de l'architecture LSTM introduite par Hochreiter et al. [Hochreiter and Schmidhuber, 1997], chaque bloc de mémoire comportait deux portes multiplicatives, une pour l'entrée et une pour la sortie. Le rôle de ces portes est, d'une part, protéger le contenu du CEC des activations provenant des nouvelles entrées (pour le cas des portes d'entrée), et d'autre part, protéger le reste du réseau du contenu du CEC si celui-ci n'est pas pertinent (pour le cas des portes de sortie).

D'autres améliorations de l'architecture LSTM ont depuis été présentées. [Cho *et al.*, 2014] ont proposé l'introduction d'une porte multiplicative supplémentaire (appelée porte d'oubli) afin de permettre à la cellule de remettre à zéro le contenu du CEC. Une autre amélioration proposée par [Cho *et al.*, 2014] est l'utilisation des peepholes, qui sont des connections supplémentaires entre le CEC et les différentes portes multiplicatives, qui permettent à ces derniers

d'espionner le contenu du CEC, leur donnant ainsi accès à des informations supplémentaires lors de leurs ouvertures et fermetures.

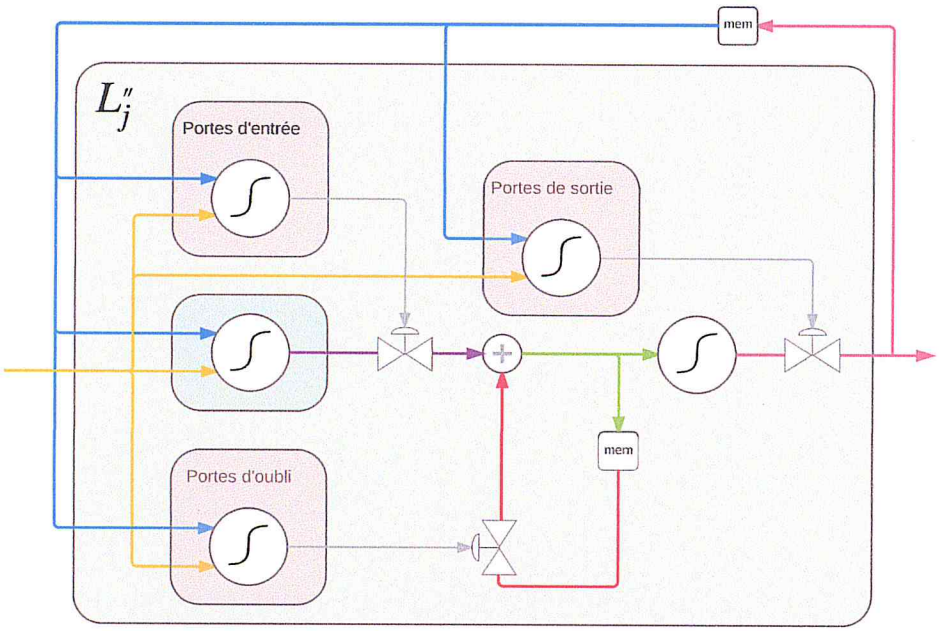


FIGURE 2.14 – Visualisation synthétique de la propagation de l'information lors de la passe avant dans une couche LSTM [Gelly, 2017]

La figure 2.14 montre une visualisation synthétique de la propagation de l'information lors de la passe avant dans une couche LSTM (sans *"peepholes"* pour faciliter la lisibilité). Les 3 portes agissent directement sur les vannes qui contrôlent le débit d'information qui provient de l'entrée, celui qui provient de la mémoire interne, mais également l'information qui sort de la couche. [Gelly, 2017]

2.4.4.2 Réseau récurrent à portes (ou GRU pour Gated Recurrent Unit)

Il est important de comprendre que la rétro-propagation du gradient classique (vu dans 2.4.1) est moins performante lorsqu'on ajoute des couches. En effet, l'algorithme est tel que plus une couche est éloignée de la couche de sortie, moins les poids de ses connexions seront modifiés durant la dernière étape de la rétro-propagation du gradient, ce phénomène est souvent désigné sous le terme de la disparition du gradient (en anglais *vanishing gradient*) [Mioulet, 2015], les GRU vise à résoudre ce problème qui se pose avec les réseaux récurrent standard, introduit par [Cho, et al. 2014], et considéré comme une variante du LSTM car les deux sont

conçus de manière similaire et, dans certains cas, produisent des résultats tout aussi excellents. (voir figure 2.15)

Afin de résoudre le problème de disparition du gradient (*vanishing gradient*) d'un RNN standard, le GRU utilise ce que l'on appelle une porte de *mise à jour* (Update gate)et une porte de *réinitialisation* (Reset gate).

- La porte de mise à jour (*update gate*) aide le modèle à déterminer quelle quantité d'informations passées (provenant des étapes précédentes) doit être transmise à l'avenir. C'est vraiment puissant parce que le modèle peut décider de copier toutes les informations du passé et éliminer le risque de problème de disparition du gradient.
- La porte de réinitialisation (*reset gate*) : Essentiellement, cette porte est utilisée à partir du modèle pour décider quelle quantité d'informations à oublier.

D'une autre manière, ce sont deux vecteurs qui décident quelles informations doivent être transmises à la sortie. Leurs particularités résident sur le fait qu'il peuvent être entraînée pour garder l'information depuis un certain temps, sans perte d'informations qui ne sont pas pertinentes à la prédiction.

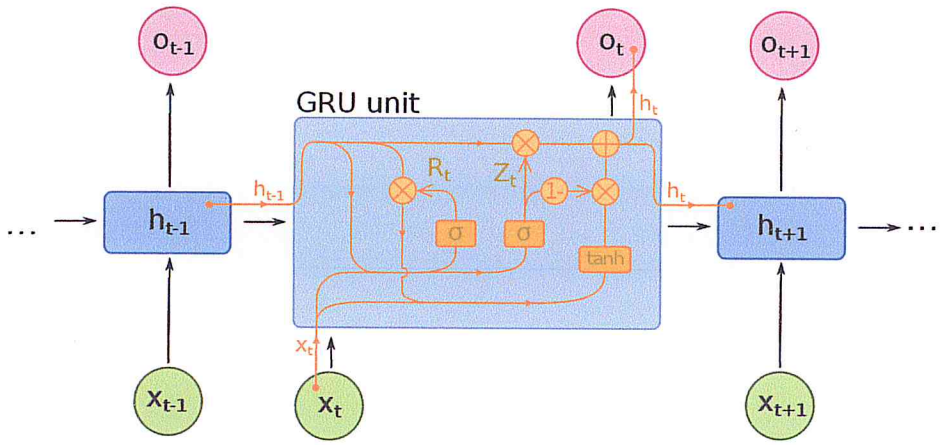


FIGURE 2.15 – Réseau récurrent à portes (GRU)

2.4.5 Réseaux antagonistes génératifs (generative adversarial networks ou GANs)

Generative adversarial networks (ou GANs) sont l'un des algorithmes de Machine Learning les plus populaires développés ces derniers temps, datant de 2014 inspirée de la théorie des

jeux pour entraîner conjointement deux réseaux antagonistes : un générateur G et un discriminateur D (comme le montre la figure 2.16) , en concurrence les uns contre les autres. Le générateur fait passer de fausses données (fake data) au discriminateur. Le discriminateur voit également des données réelles et prédit si les données qu'il reçoit sont réelles ou fausses. Le générateur est formé pour tromper le discriminateur, il veut produire des données qui ressemblent le plus possible à des données réelles. Et le discriminateur est formé pour comprendre quelles données sont réelles et lesquelles sont fausses. Ce qui finit par arriver, c'est que le générateur apprend à rendre les données indiscernables des données réelles au discriminateur. Le premier réseau va donc devoir tromper le deuxième : ils sont en opposition. [Goodfellow *et al.*, 2014]

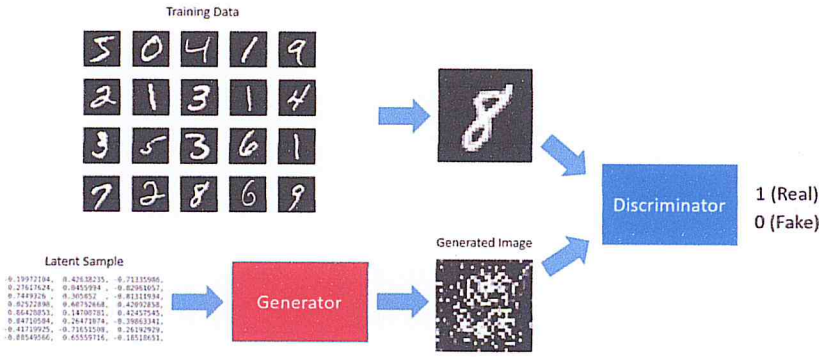


FIGURE 2.16 – Generative Adversarial Network (GAN) [Shibuya, 2017]

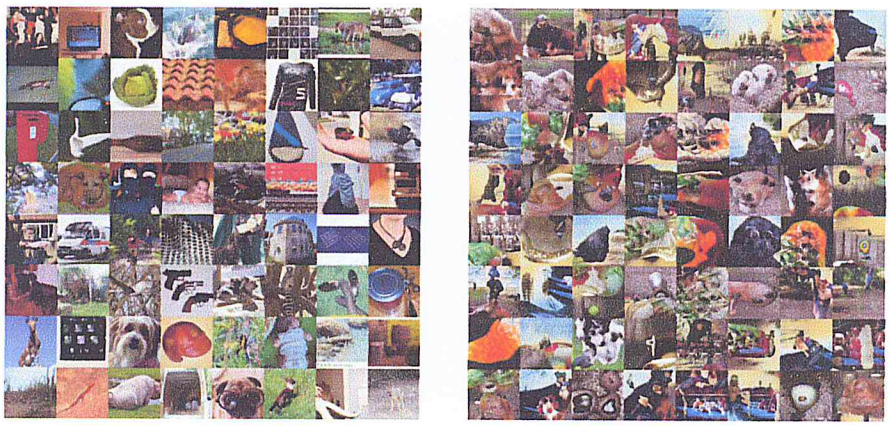


FIGURE 2.17 – à gauche : des images réelle (à partir d'Imagenet), à droite : images générés par le réseau génératif [OpenAI, 2016]

2.4.6 Les Autoencodeurs

Un auto-encodeur, ou auto-associateur [Bengio, 2009] est un réseau de neurones artificiels utilisé pour l'apprentissage non supervisé de caractéristiques discriminantes [Liou *et al.*, 2014]. L'objectif d'un auto-encodeur est d'apprendre une représentation (encodage) d'un ensemble de données, généralement dans le but de réduire la dimension de cet ensemble. Récemment, le concept d'auto-encodeur est devenu plus largement utilisé pour l'apprentissage de modèles génératifs [Kingma and Welling, 2013]

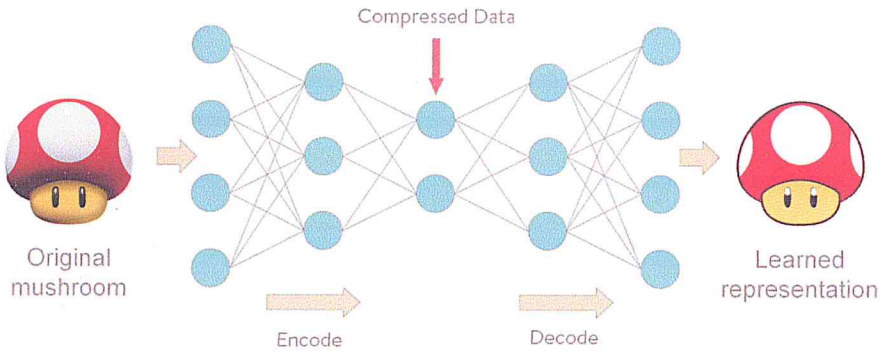


FIGURE 2.18 – Illustration simplifié d'un AutoEncoder [science, 2017]

La forme la plus simple d'un auto-encodeur est un réseau de neurones non récurrents qui se propage vers l'avant, très semblable au réseau neurone multicouches ayant une couche d'entrée, une couche de sortie ainsi qu'une ou plusieurs couches cachées les reliant, mais avec toutefois une couche de sortie possédant le même nombre de nœuds que la couche d'entrée, son objectif étant de reconstruire ses entrées (plutôt que de prédire une valeur cible Y étant donné les entrées X). Par conséquent, un auto-encodeur est un modèle d'apprentissage non supervisé. [Liou *et al.*, 2008] Alors que les Autoencodeurs peuvent être utilisés pour la compression, leur performance est évidemment à perte, en raison de sa réduction de dimensionnalité inhérente.

Une utilisation plus appropriée pour un Autoencodeur est le débruitage d'image (figure 2.19). Un DAE (Denoising Autoencoder) Le principe du débruitage avec un DAE (pour Denoising Autoencoder) est de pouvoir reconstruire des données à partir d'une entrée de données corrompues. Après avoir donné à l'autoencodeur les données corrompues, nous forçons la couche cachée à apprendre seulement les fonctionnalités les plus robustes. La sortie sera alors

une version plus raffinée des données d'entrée [Collis, 2017] comme le montre la figure [OpenDeep, 2015]

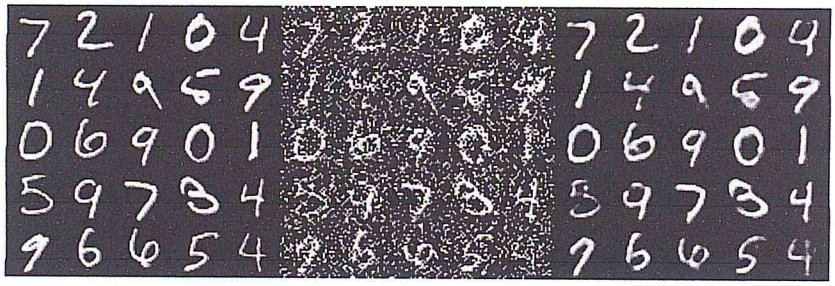


FIGURE 2.19 – Exemple d'un DAE sur des chiffres manuscrits, à gauche : images originales, au milieu : des images d'entrée avec bruits, à droite : images restaurés (bruit enlevé) [OpenDeep, 2015]

2.4.6.1 Variational Auto encoder (VAE)

Les autoencodeurs variationnel (VAE) ont été développés par [Kingma and Welling, 2013] comme l'une des approches les plus utiles à l'apprentissage de la représentation de données complexes au cours des dernières années. Les VAEs ont déjà démontré des performances prometteuses dans des données complexes, notamment des chiffres manuscrits, des visages, des numéros de maison, des modèles de discours et physiques. Ils ont la même structure des autoencodeurs comprenant des encodeurs, des décodeurs et des couches cachées (latentes)

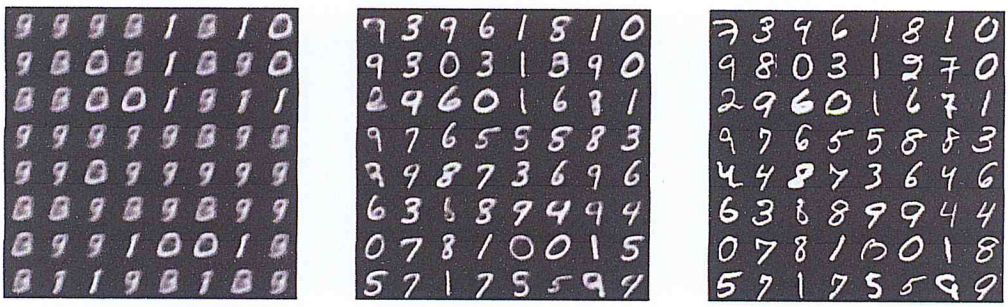


FIGURE 2.20 – Reconstruction des chiffres manuscrites avec les VAE , à droite : données originales, à gauche : après une seule itération dans la phase d'entraînement, au milieu : 9^{eme} itération. [Frans, 2016]

Le principale avantage des VAEs par rapport aux autoencodeurs traditionnels est le fait d'apprendre la vraie distribution des données d'entraînement plutôt que de simplement se sou-

venir de l'ensemble de données d'entraînement particulier, ce qui permet d'améliorer considérablement la performance de la représentation. [Nezhad *et al.*, 2018] Soit en utilisant des VAEs, il est non seulement possible de compresser des données mais également possible de générer de nouveaux objets du type que l'autoencoder a vu auparavant.

FIGURE 2.21 – Reconstruction d'images par VAE prises à partir d'un ensemble de test séparé [Larsen *et al.*, 2015]

2.5 Conclusion

Dans ce chapitre, nous avons donné une vue d'ensemble sur l'apprentissage automatique, ainsi que sur ses différents type (apprentissage supervisé, apprentissage non supervisé, apprentissage semi-supervisé et apprentissage par renforcement). Nous avons réalisé aussi une étude détaillée sur les réseaux de neurones vus leur intérêt récent par la communauté scientifique. Ils ont pu donner des résultats performants pour les taches de traitement d'images et traitement vidéo.

Dans le prochaine chapitre, nous détaillerons les travaux réalisés dans le cadre de la génération des résumés vidéos.

CHAPITRE 3

TRAVAUX PROPOSÉ POUR LA GÉNÉRATION DES RÉSUMÉS VIDÉOS

Sommaire

3.1 Introduction 39

3.2 Travaux basés sur une seule source vidéo 39

3.2.1 Méthodes basées sur l'échantillonnage 39

3.2.2 Méthodes basées sur les plans 40

3.2.3 Méthodes basées sur les scènes (ou macro-segments) 41

3.2.4 Approches basées sur l'extraction d'événements intéressants 42

3.3 Travaux basé sur plusieurs sources vidéos 42

3.3.1 Approche basé sur la théorie des graphes : 43

3.3.2 Approches basées sur les caractéristiques spatio-temporelles 44

3.4 Conclusion 45

3.1 Introduction

Poussés par la croissance exponentielle du nombre de vidéos en ligne au cours des dernières années, les recherches sur les résumés vidéos ont attiré de plus en plus d'attention, ce qui a conduit à diverses méthodes proposées dans la littérature pour faciliter la navigation vidéo à grande échelle.

Dans les travaux existants [Money and Agius, 2008] [Rajendra, 2014] , les résumés de vidéos peuvent être de deux types. Le premier consiste à extraire un ensemble d'images, le deuxième consiste à créer une nouvelle vidéo à partir de la vidéo d'entrée. Le premier type, généralement appelé résumé statique de la vidéo, est une collection d'images représentatives qui sont soigneusement extraites à partir de la vidéo. Ces images sont appelées images-clés. Chacune d'entre elles représente le contenu visuel d'une partie de la vidéo. La visualisation de ce type de résumé est rapide et la complexité de son algorithme de création est généralement faible. Le second type, également connu sous le nom de résumé dynamique, est un ensemble de segments vidéo sélectionnés à partir de la vidéo d'entrée. Ce type de résumés conserve les propriétés dynamiques de la vidéo d'entrée et par conséquent il est plus agréable à regarder qu'un résumé statique. Il est également plus expressif, car il comprend à la fois de l'information visuelle et audio. Mis à part quelques cas d'utilisation spécifiques où la sélection d'un ensemble d'images représentatives est suffisante, un résumé sous forme d'une vidéo est généralement plus utile dans la pratique. [Boukadida, 2015]

Dans ce chapitre, nous décrivons les différents travaux existants de création de résumés vidéo regroupés en deux catégories : ceux exploitant une seule source vidéo, et ceux utilisant plusieurs sources vidéo.

3.2 Travaux basés sur une seule source vidéo

3.2.1 Méthodes basées sur l'échantillonnage

Les premiers travaux [Mills *et al.*, 1992] [Taniguchi *et al.*, 1995] concernant les résumés de vidéos consistent à choisir les images clés en sous-échantillonnant uniformément ou aléatoirement la séquence originale. L'inconvénient des méthodes qui n'étudient pas le contenu, est

la non représentation de certaines parties de la vidéo et la possible redondance de certaines images clés avec des contenus similaires.

3.2.2 Méthodes basées sur les plans

Des travaux plus élaborés tentent d'extraire des images clés en s'adaptant au contenu de la vidéo. La détection des plans est réalisée pour mieux ajuster la sélection des images clés au contenu de la vidéo. Une façon simple pour représenter les différents plans de la vidéo est d'extraire la première image du plan comme image clé [Nagasaka and Tanaka, 1991] [Tonomura *et al.*, 1993] ou les première et dernière images du plan [Ueda *et al.*, 1991]. Ces méthodes semblent efficaces pour décrire les plans stationnaires où le contenu varie peu. En revanche, elles ne fournissent pas une représentation satisfaisante pour les plans avec de forts mouvements de caméra.

D'autres travaux choisissent alors de représenter le contenu des vidéos en employant des caractéristiques visuelles de bas niveau comme la couleur, le mouvement ou la texture. Dans [Zhang *et al.*, 1995] [Bilge Gunsel, 1997] , le nombre d'images clés dépend du contenu des plans. La première image du plan est sélectionnée comme image clé. Puis, si la distance entre l'histogramme couleur de la dernière image clé sélectionnée et l'image courante est supérieure à un seuil alors une nouvelle image clé est choisie. La sélection de la première image n'est pas forcément judicieuse puisqu'elle peut être soumise aux effets de transition (fondu enchaîné) entre les plans.

L'approche décrite dans [Zhuang *et al.*, 1998] compare les images d'un plan suivant leurs histogrammes de couleur puis réalise le rassemblement des images en plusieurs groupes. Seuls les groupes de taille assez importante sont conservés et les images les plus proches du centre de gravité de chaque groupe sont alors choisies comme images clés.

L'inconvénient de travailler au niveau des plans est que le nombre d'images clés peut être trop important pour représenter la vidéo. [Guironnet, 2006]

3.2.3 Méthodes basées sur les scènes (ou macro-segments)

Des travaux ont été réalisés en ne considérant pas comme unité de la vidéo le plan mais en définissant des unités de plus haut niveau. Suivant les auteurs, différents niveaux de hiérarchie sont étudiés pour créer le résumé de vidéo. Par exemple, le regroupement de plans par similarité peut être considéré comme un niveau plus élevé que la segmentation en plans et aura pour conséquence de sélectionner moins d'images clés

Des méthodes de résumé ont été conçues pour décrire seulement les plans les plus représentatifs de la vidéo. L'approche décrite dans [Uchihashi *et al.*, 1999] consiste d'abord à assigner un groupe à chaque image, puis à réunir les deux groupes les plus similaires de manière itérative. L'algorithme s'arrête quand la similarité entre les groupes est inférieure à un seuil. Pour chaque groupe créé, les images adjacentes sont déterminées et constituent un segment. Une mesure d'importance est alors calculée pour chaque segment suivant sa longueur et sa rareté. Tous les segments de petite taille sont alors supprimés et seuls les segments ayant une mesure d'importance suffisante sont conservés. Puis, l'image la plus proche du centre de gravité de chaque segment est extraite comme image clé. Le résumé est ensuite présenté en ajustant la taille des images clés suivant leur importance.

Un algorithme pour organiser les images clés selon leur taille a été proposé dans [Girgensohn, 2003] afin d'optimiser l'affichage du résumé. [Sun and Kankanhalli, 2000] ont proposé une méthode de résumé où la vidéo est divisée uniformément en segments. A chaque segment est associée une mesure de changement qui est égale à la distance entre les première et dernière images. Les différentes mesures sont alors ordonnées puis séparées en 2 groupes : un groupe de petit changement et un groupe de grand changement. Pour le groupe de petit changement, les première et dernière images du segment sont conservées alors que pour le groupe de grand changement, toutes les images sont gardées. Finalement si le nombre d'images clés désiré est atteint, l'algorithme s'arrête sinon les images conservées constituent une nouvelle vidéo et celle-ci est à nouveau divisée en segments jusqu'à atteindre le nombre désiré d'images clés.

Dans [Kopf *et al.*, 2004], une mesure de pertinence est associée à chaque plan suivant différentes caractéristiques (détection des visages, identification d'objets en mouvement,. . .) et

les plans sont sélectionnés dans l'ordre décroissant de cette mesure jusqu'à atteindre la taille du résumé souhaitée. L'inconvénient des approches qui travaillent sur la totalité de la vidéo est qu'elles peuvent s'avérer très contraignantes par le temps de calcul et la mémoire sollicitée.

3.2.4 Approches basées sur l'extraction d'événements intéressants

Une autre façon de créer des résumés vidéo est de détecter les moments forts (appelés en anglais *highlights*) de la vidéo d'entrée.

L'un des premiers travaux utilisant ce principe est le projet Informedia [Smith and Kanade, 1995]. ce projet se base sur la détection de caractéristiques audiovisuelles de bas niveau. Un résumé vidéo est créé en utilisant la détection de visages, la détection de texte et la détection du mouvement de la caméra. Ce résumé est créé en utilisant un système de classement qui considère que les images présentant les visages ou le texte sont les plus importantes, les images statiques qui suivent le mouvement de la caméra sont moins importantes... etc. Bien que l'intégration de l'information visuelle, audio et textuelle soit la meilleure façon de comprendre une vidéo, la génération de résumés basés sur une telle technique nécessite encore une intervention manuelle.

Un nouvel algorithme de classification a été proposé dans [Peng and Ngo, 2005]. Cet algorithme utilise une approche de clustering basée sur l'utilisation de graphes non orientés mesurant la similarité visuelle entre toutes les paires d'événements d'une vidéo. Les moments forts sont détectés en sélectionnant le clip représentatif de chaque cluster et en se basant sur les propriétés des clusters : la taille du cluster et la globalité d'un événement.

La création de résumés basés sur la détection des événements intéressants est une tâche difficile dans l'absence d'une connaissance a priori du type de la vidéo en question. [Guironnet, 2006]

3.3 Travaux basé sur plusieurs sources vidéos

La plupart des travaux actuels se focalisent généralement sur la construction du résumé d'une seule vidéo, seuls quelques-uns se sont portés au problème de résumés multi-vidéos où la prise en compte d'autres contraintes et éléments s'impose. Nous citons par exemple le fait que

plusieurs informations (scènes, personnes) sont présentes telles qu'elles sont ou d'une façon similaire dans diverses vidéos. La construction de résumés vidéos indépendamment les uns des autres peut provoquer une présence d'informations redondantes dans les résumés créés.

Dans ce qui suit, nous présentons quelques travaux faits sur la construction de résumés multi-sources.

3.3.1 Approche basé sur la théorie des graphes :

[Fu *et al.*, 2010] leur travail consiste à construire un graphe des shots spatio temporelle et de formuler le problème du résumé en tant que tâche d'étiquetage du graphe. Une telle représentation donne la possibilité de résoudre le problème de résumé multi-vues en utilisant la théorie des graphes. (voir figure 3.1)

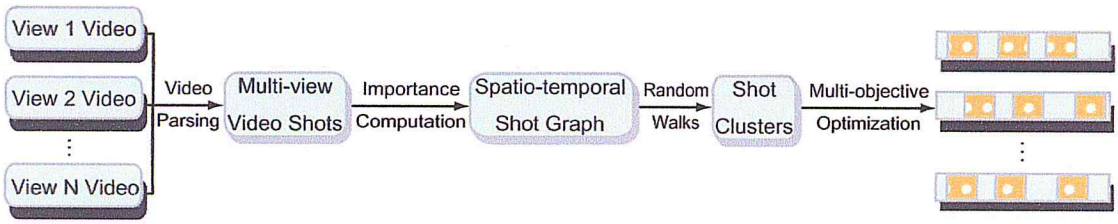


FIGURE 3.1 – Un aperçu du résumé vidéo multi-vue d'après [Fu *et al.*, 2010]

Un graphe spatio-temporel est utilisé pour la représentation de vidéos multi-vues. Le graphe de shots est dérivé d'un hypergraphe qui intègre différentes corrélations entre les prises de vues dans chaque vidéo ainsi que sur plusieurs vidéos.

Les Random Walks sont utilisés pour regrouper les clusters centrés sur les événements et le résumé final est généré par une optimisation multi-objective. L'optimisation multi-objective peut être configurée de manière flexible pour répondre aux différentes exigences du résumé.

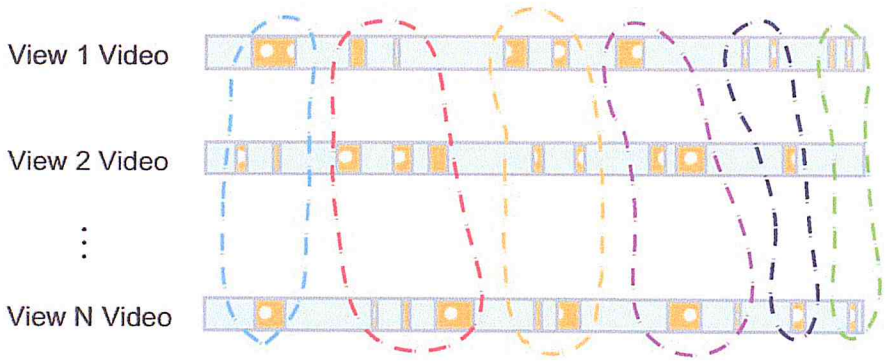


FIGURE 3.2 – Représentation d'événements similaires généré par les random walks sous forme de clusters (entourés de cercles en pointillés) [Fu *et al.*, 2010]

Le storyboard vidéo multi-vues et le forum événementiel sont présentés pour représenter le résumé vidéo multi-vues. Le storyboard reflète naturellement les corrélations entre les résumés à plusieurs vues qui décrivent le même événement important. Le tableau des événements réunit en série des images multi-vues centrées sur l'événement dans l'ordre temporel. Avec le forum d'événements, un résumé vidéo unique qui facilite La navigation rapide de la vidéo résumée peut être facilement généré. Cependant, les mêmes auteurs [Fu *et al.*, 2010] ont mentionné que le processus de création de graphiques est intrinsèquement complexe et qu'il consomme la majeure partie du temps de traitement global. De plus, de courtes marches (*short walks*) peuvent se produire, et par conséquent engendré des résultats erronés de clustering.

3.3.2 Approches basées sur les caractéristiques spatio-temporelles

Une méthode de résumé statique multi-vues proposé par [Panda *et al.*, 2016] qui extrait un ensemble d'images clés pour présenter les parties les plus importantes des vidéos d'entrée sous forme de story-boards (comme montré dans la figure 3.3) .

FIGURE 3.3 – Quelques événements résumés sur le dataset Office Lobby produit par [Panda *et al.*, 2016]

Bien que les images clés soient un moyen utile d'indexer les vidéos, elles sont limitées dans la mesure où toute l'information de mouvement est perdue. Cela limite leur utilisation dans

de nombreuses applications tel que la vidéo surveillance où le résumé dynamique (*video skimming*) semble être mieux adapté pour obtenir des informations significatives en peu de temps. Les mêmes auteurs [Panda *et al.*, 2016] ont proposé une améliorations dans [Panda and Roy-Chowdhury, 2017] où ils proposent un résumé dynamique multi-vues à base de plans, Pour ce faire, ils utilisent un schéma de représentation vidéo basé sur les caractéristiques spatio-temporelles C3D (que nous détaillerons dans le prochain chapitre), Ensuite, ils exploitent les corrélations multi-vues sans supposer de correspondance préalable entre les vidéos. Plus précisément, ils formulent la tâche de trouver des résumés sous la forme d'un problème d'optimisation qui mesure l'importance des plans. Enfin, l'approche produit un résumé vidéo composé des plans ayant le score d'importance le plus élevé.

3.4 Conclusion

Dans ce chapitre, nous avons présenté des méthodes existantes de création de résumés. Le but de cette étude est de situer notre travail par rapport à quelques travaux déjà efectués dans ce domaine. Cette étude a révélé aussi qu'il y a peu de travaux consacrés au cas des résumés multi-vidéos. La construction de résumés vidéos indépendamment les uns des autres peut provoquer une présence d'informations redondantes dans les résumés créés. Des méthodes spécifiques doivent être conçues afin de prendre en considération ces similarités et produire un ensemble de résumés plus eficaces.

CHAPITRE 4

APPROCHE PROPOSÉ

Sommaire

4.1 Introduction 47

4.2 Vue globale de l'approche 48

4.3 Phase de pré-traitement 49

4.4 Phase d'extraction de caractéristiques 50

4.4.1 Le modèle C3D (Conv3D) 51

4.4.2 Architecture du modèle C3D 52

4.5 Phase de réduction de dimension 53

4.6 Création du résumé 54

4.7 Conclusion 57

4.1 Introduction

Les réseaux de caméras de surveillance sont partout de nos jours. Le volume de données collectées par un tel réseau de capteurs de vision déployés dans de nombreux contextes allant des besoins de sécurité à la surveillance de l'environnement ce qui répond clairement aux exigences des grandes masse de données. (voir figure 4.1)

Les difficultés d'analyse et le traitement de ces grandes données vidéo sont évidentes chaque fois qu'il y a un incident qui nécessite de fouiller dans de vastes archives vidéo pour identifier les événements d'intérêt. [Panda and Roy-Chowdhury, 2017]

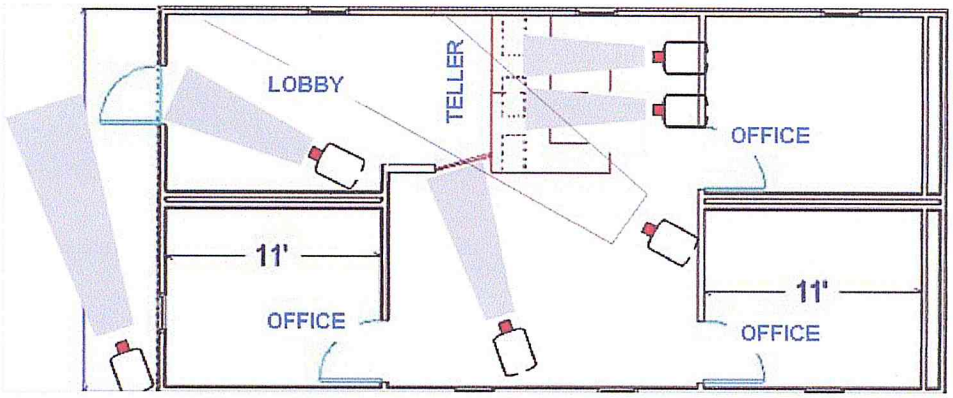


FIGURE 4.1 – Illustration d'un réseau de caméra multi vues [Cisco, 2016]

En conséquence, le résumé vidéo, qui extrait automatiquement un résumé bref mais informatif de ces vidéos a attiré une attention intense au cours des dernières années. Bien que le résumé vidéo ait fait l'objet d'études approfondies au cours des dernières années, beaucoup de méthodes concentré sur l'élaboration d'un résumé vidéo a une vue unique. Cependant, un autre problème important et rarement abordé dans ce contexte est de trouver un résumé informatif à partir de plusieurs vues. (comme le montre la figure 4.2)

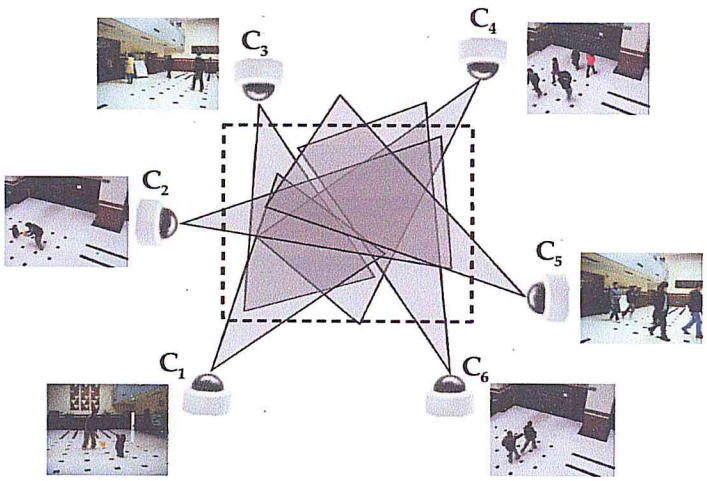


FIGURE 4.2 – Une illustration d'un réseau de caméras multi-sources où six caméras C1, C2, C3, C4, C5, C6 observe une zone (rectangle noir) à partir de différents angles. [Panda *et al.*, 2016]

Une solution à ce problème est donc de créer automatiquement un résumé de la vidéo. Le résumé vidéo permet de répondre à ce besoin en fournissant un aperçu général et rapide de l'ensemble du contenu audiovisuel de la vidéo originale et en présentant les parties intéressantes pour l'être humain.

À travers ce quatrième chapitre nous allons présenter, et expliquer en détail, les principes de fonctionnement de notre approche utilisée pour la création automatique de résumé vidéo tiré de plusieurs vues d'une même scène et qui est basée sur l'apprentissage profond (*deep learning*)

4.2 Vue globale de l'approche

Un aperçu global de notre approche pour la génération du résumé vidéo est illustré dans la figure ci dessous.

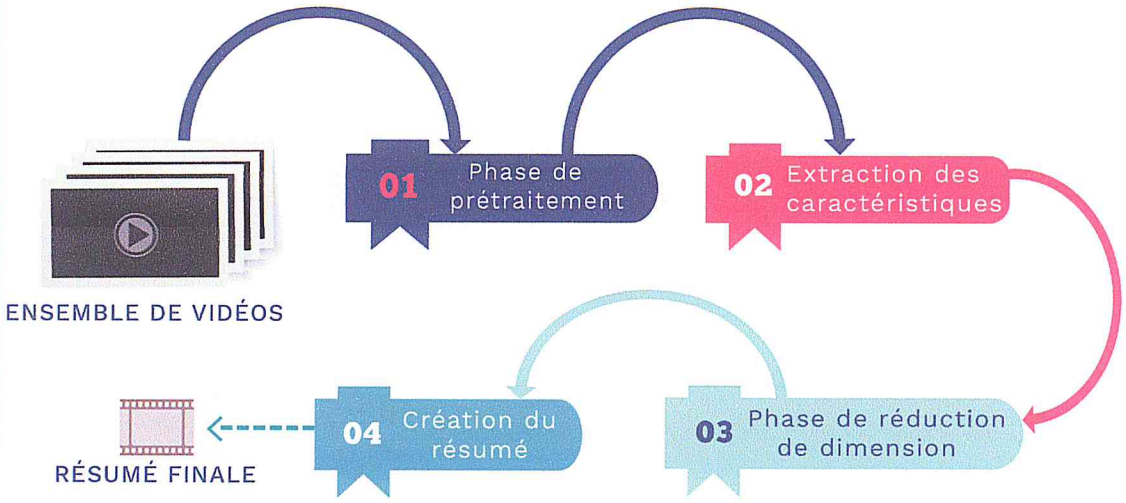


FIGURE 4.3 – Schéma globale de notre approche

Nous nous concentrons sur le développement d'une approche non supervisée pour la sélection d'un sous ensemble de plans (séquences) qui constituent le résumé multi vues. Dans ce qui suit nous décrivons les différentes phases que constitue notre approche :

1. Phase de pré-traitement
2. Phase d'extraction des caractéristiques profondes
3. Phase de réduction de dimension
4. Création du résumé

4.3 Phase de pré-traitement

Dans notre approche, une étape de prétraitement des vidéos d'entrées est nécessaire pour la création automatique du résumé. Cette phase consiste à l'extraction des frames de chaque vidéos, puis de les redimensionner. Cette phase permet d'optimiser le temps de calcul. (la figure 4.4 illustre ces différentes étapes)

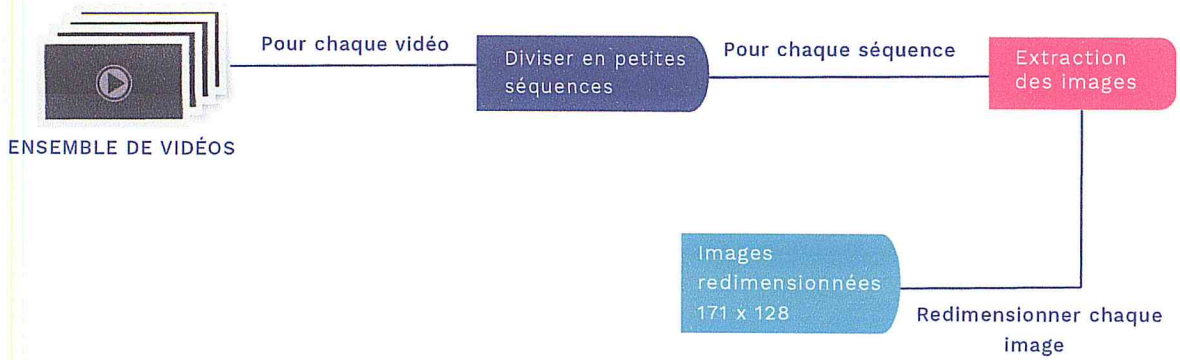


FIGURE 4.4 – Schéma globale de la phase prétraitement

1. Pour chaque vidéo i parmis l'ensemble des vidéos d'entrée
2. Ségmentation de la vidéo en petite séquence d'une durée déterminé (dans notre étude, nous avons choisis une durée t = 20 secondes)
3. Extraction des frames de chaques segments vidéos
4. Redimensionnement des frames de chaques segments vidéos (dans notre étude, nous avons choisi les résolutions 171 x 128)

À noter que le résumé final sera construit en rassemblant les segments vidéo les plus représentatifs extraits de la source vidéo (contenant les images-clés)

4.4 Phase d'extraction de caractéristiques

Cette phase consiste à extraire, analyser et de caractériser l'information brute présente sur les images des segments vidéos afin d'identifier les éléments qui les constituent. Le rôle principal de ces caractéristiques en vision par ordinateur est de transformer l'information visuelle sous formes de vecteurs caractéristiques (comme le montre la figure 4.5)

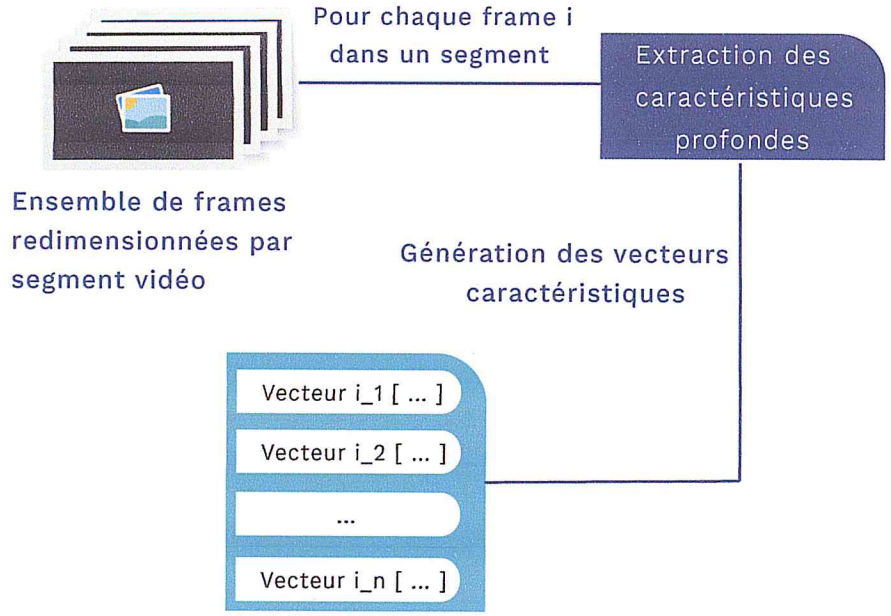


FIGURE 4.5 – Schéma globale de la phase d'extraction des caractéristiques profondes

Afin de construire ces vecteurs caractéristiques des vidéos, nous avons opté pour une architecture profonde obtenue en entraînant un réseau à convolution tridimensionnel (C3D pour Conv3D) sur une grande collection de vidéos annotées (nommé *Sports-1M*), la plus large qui existe actuellement avec *1,1* million de vidéos et *487* catégories. [Tran *et al.*, 2014] Dans ce qui suit nous détaillons l'architecture C3D et ses avantages dans notre cas de création de résumé vidéo.

4.4.1 Le modèle C3D (Conv3D)

Le modèle C3D est une approche simple, mais efficace pour l'apprentissage des caractéristiques spatio-temporelles (voir figure 4.6) , capable de modéliser simultanément l'apparence et les informations de mouvement et surpasse les fonctionnalités 2D des ConvNet sur diverses tâches de classification de vidéos.

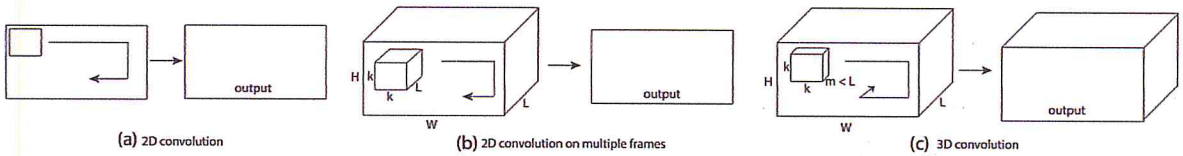


FIGURE 4.6 – Différence entre l'opération de convolution 2D et convolution 3D [Tran *et al.*, 2014]

La figure 4.6 compare la différence entre, **(a)** la convolution 2D sur une seule image, **(b)** la

convolution 2D sur plusieurs images et **(c)** la convolution 3D sur plusieurs images. La convolution 3D préserve l'information temporelle et la transmet à la couche suivante tandis qu'elle n'est pas préservé dans la convolution 2D.

Parmi les avantages du modèle C3D c'est qu'il est générique, c'est-à-dire qu'il permet d'obtenir des résultats performants en matière de reconnaissance d'objets, de classification de scènes, d'étiquetage et de similarité d'actions dans les vidéos.

Le modèle C3D permet aussi d'obtenir de meilleures précisions que les descripteurs tel que SIFT ou SURF (*vu en chapitre I*) et plus rapide en calcul (**91** fois plus rapide que SIFT ou SURF et **deux** fois plus rapide que les méthodes actuelles de classification basée sur l'apprentissage profond) [Tran *et al.*, 2014]

4.4.2 Architecture du modèle C3D

[Tran *et al.*, 2014] ont constaté qu'une configuration homogène avec des filtres de convolution de taille 3 × 3 × 3 est la meilleure option pour les ConvNets 3D. Cette constatation est également cohérente avec une constatation similaire dans ConvNets 2D [Simonyan and Zisserman, 2014].

Les ConvNet 3D possède 8 couches de convolution, 5 couches de pooling, suivies de deux couches entièrement connectées, et une couche de sortie softmax. L'architecture du réseau est présentée à la figure 4.7

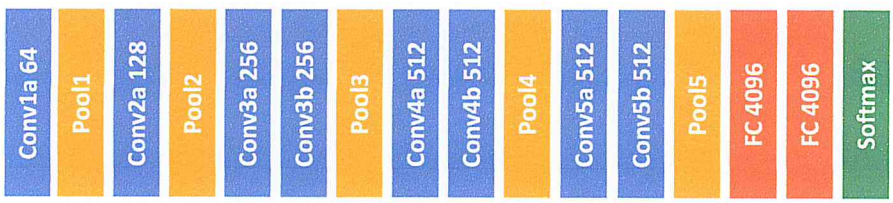


FIGURE 4.7 – Architecture du ConvNet3D [Tran *et al.*, 2014]

Tous les filtres de convolution 3D sont de taille 3 × 3 × 3 (longueur × hauteur × largeur) avec un stride de 1 × 1 × 1 . Le nombre de filtres est indiqué dans chaque case. Les couches de pooling 3D sont notées de pool1 à pool5. Tous les filtres de pooling sont de taille 2 × 2 × 2, sauf pour pool1 est de 1 × 2 × 2. Chaque couche entièrement connectée possèdent 4096 unités de sortie.

Une méthode de déconvolution a été utilisé dans [Zeiler and Fergus, 2013] pour visualiser les caractéristiques apprises par le C3D dans certaines couches internes (comme le montre les figure 4.8 et 4.9)

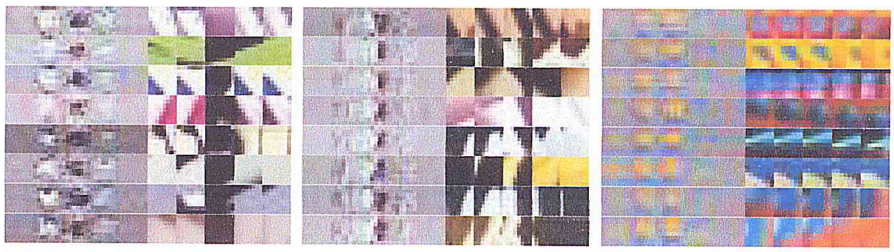


FIGURE 4.8 – Déconvolution des caractéristiques apprises par le C3D sur la couche conv2a : Les filtres appris détectent les changements de prise de vue, les changements d'orientation des bords et les changements de couleur. [2] [Tran *et al.*, 2014]

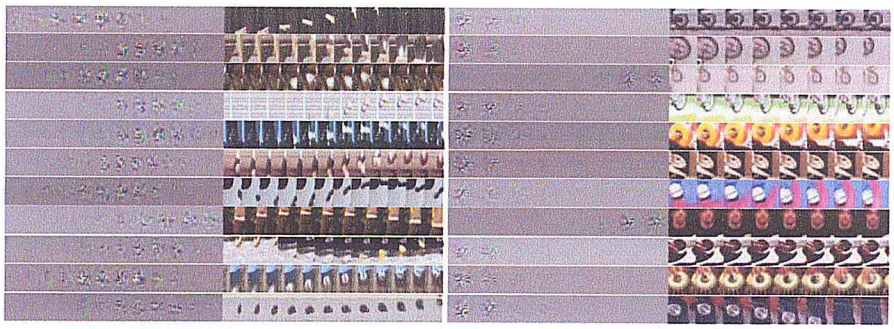


FIGURE 4.9 – Déconvolution des caractéristiques apprises par le C3D sur la couche conv3b : Les filtres détectent les trajectoires d'objets et les objets circulaires [Tran *et al.*, 2014]

Pour notre cas, On extraira les valeurs caractéristiques trouvées à la sixième couche entièrement connecté (FC6)

4.5 Phase de réduction de dimension

Une fois que toutes les caractéristiques profondes sont extraites avec le modèle C3D, on voit bien que la dimension de chaque vecteur est égale à 4096, qui rend le temps de calcul long. Pour cette raison, nous avons vu que la réduction de dimension de l'espace est nécessaire. Nous avons opté pour l'utilisation des autoencoders, qui sont similaires aux techniques de réduction de la dimensionnalité tel que l'analyse des composants principaux PCA [Comon, 1994] ou l'algorithme t-SNE pour Stochastic Neighbor Embedding [van der Maaten *et al.*, 2008]. Ils créent

un espace où les parties essentielles des données sont préservées, tandis que les parties non essentielles (ou bruyantes) sont supprimées.

Un autoencodeur se compose de deux parties, à savoir un encodeur et un décodeur (comme le montre la figure 4.10). Alors que l'encodeur vise à compresser les données d'entrée originales en une représentation de faible dimension, le décodeur tente de reconstruire les données d'entrée originales sur la base de la représentation de faible dimension générée par l'encodeur. Par conséquent, l'autoencodeur a été largement utilisé pour supprimer le bruit de données ou comme dans notre cas, afin de réduire la dimension des données.

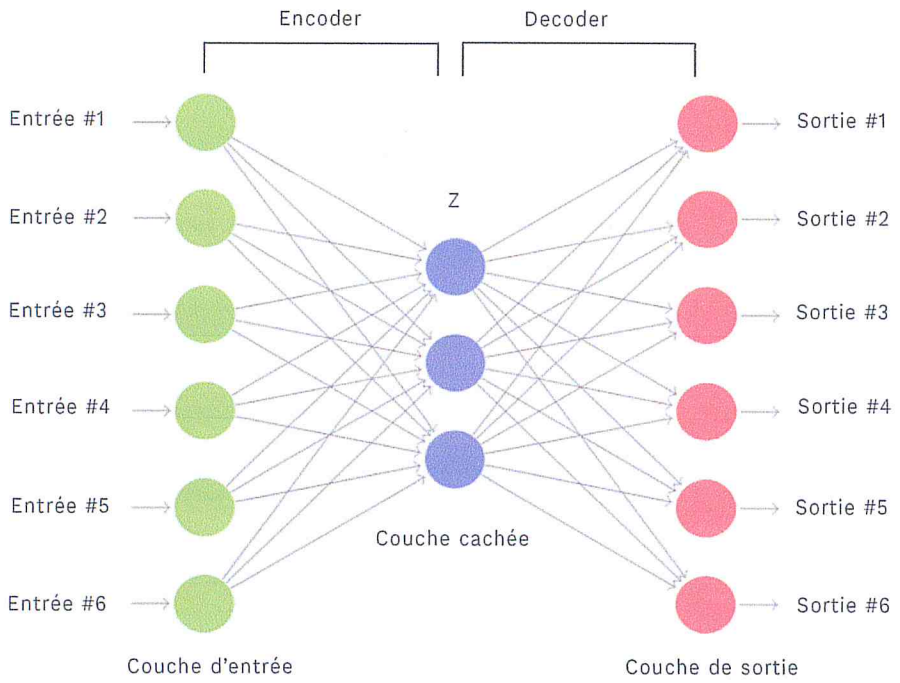


FIGURE 4.10 – Représentation d'un simple autoencodeur [fouché, 2017]

Dans notre cas, l'entrée de l'autoencoder correspond au vecteur caractéristique extraite à partir du modèle C3D de dimension 4096, nous extrayons les données compressé notée Z de dimension réduite à partir de la couche cachée (qui correspond à la sortie de l'encoder).

4.6 Création du résumé

Après la réduction de dimension, on passe à l'étape suivante qui consiste en la production du résumé final des vidéos originales. L'approche est basée sur les réseaux de neurones récurrents bidirectionnels (BiRNN), qui étendent le RNN unidirectionnel en introduisant

une deuxième couche cachée, où les connexions entre les deux couches cachées circulent dans l'ordre temporel opposé. Le modèle est donc capable d'exploiter l'information du passé et du futur. [Berglund *et al.*, 2015]

Plus précisément, Le BiRNN utilise la cellule de mémoire à long-court terme (LSTM) pour améliorer la capacité du RNN à capturer les dépendances dans frames des vidéos. Le réseau prédit pour chaque image une probabilité (en tant que score d'importance), à partir de laquelle une action 0,1 est réalisé pour indiquer si l'image est sélectionnée ou non dans le résumé final. Le résumé vidéo est composé d'images sélectionnées en maximisant les scores tout en veillant à ce que la longueur du résumé ne dépasse pas une limite, qui correspend généralement à 15% de la durée de la vidéo originale.

Le réseau BiRNN est entraînée en utilisant un apprentissage par renforcement (abordé en chapitre II). L'intuition derrière cette stratégie d'apprentissage est liée à la façon dont les humains résument les vidéos. En effet, l'apprentissage par renforcement vise essentiellement à optimiser le mécanisme d'action (sélection des images) d'un agent (ici il s'agit de l'algorithme) en renforçant itérativement l'agent à prendre de meilleures actions. (voir figure 4.11)

L'apprentissage par renforcement peut être schématisé de la manière suivante :

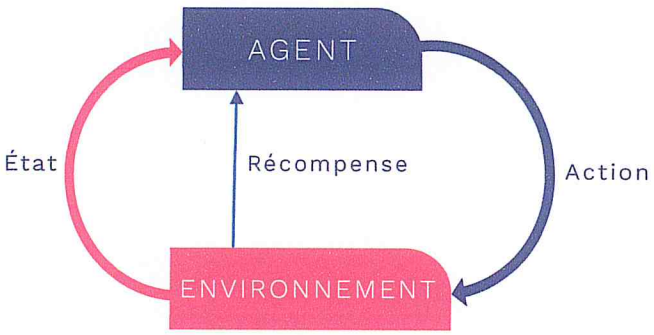


FIGURE 4.11 – Processus de décision d'un agent avec l'apprentissage par renforcement : l'agent est dans un état s, et exécute une action a. Cette action a des conséquences dans l'environnement de l'agent, et pour lequel il reçoit une récompense r, et change d'état s+1 [Microsoft, 2016]

Initialement, le système ne connaît pas les états où se trouve les récompenses, il commence donc par choisir des actions aléatoirement, il explore. Au bout d'un certain temps ou lorsqu'il a atteint un état but, le système reprend une recherche de solution à partir de l'état initial, et a chaque cycle, le système présente un comportement de moins en moins exploratoire, et de

plus en plus guidé par les qualités [Vidal and Vidal, 2006].

Dans notre cas, le réseau tente de maximiser les récompenses attendues en produisant des résumés de haute qualité. Ces récompenses sont obtenues en tenant compte de deux aspects :

1. Calcul du degré de diversité du résumé généré : en mesurant la dissimilarité entre les images sélectionnées

$$R_{div} = \frac{1}{|Y|(|Y|-1)} \sum_{t \in Y} \sum_{t' \in Y, t' \neq t}^{d} (x_t, x_{t'}) \tag{4.1}$$

Ou d s'agit de la fonction de dissimilarité calculé comme suit :

$$d(x_t, x_t') = 1 - \frac{x_t^T x_t'}{||x_t||_2 ||x_t'||_2} \tag{4.2}$$

Intuitivement, plus il y a de diversité (ou plus de dissemblable) entre les images sélectionnés dans le résumé, plus la récompense de la diversité est élevée que l'agent peut recevoir.

2. Mesurer à quel point le résumé généré peut représenter les vidéos originales, comme suit

$$R_{rep} = exp(-\frac{1}{T} \sum_{t=1}^{T} \underset{t' \in Y}{min} ||x_t - x_t'||_2) \tag{4.3}$$

Avec cette récompense, l'agent est encouragé à choisir des images qui sont proches des centres de clusters.

La création du résumé finale est traité sous forme d'un problème de sac à dos (knapsack 0/1), qui est étudié depuis plus d'un siècle mais reste un sujet très prisé de recherche. Pour résumé, il décrit comment choisir parmis un ensemble d'objets, ceux qui maximise la valeur total sans dépasser la capacité que peut supporter le sac. Dans notre cas de résumé vidéo, l'objectif est de sélectionner un maximum d'images clés, tout en respectant la taille qui ne doit pas dépasser 15% de la taille de la vidéo originale, on obtient une solution quasi optimale grâce à la programmation dynamique.

4.7 Conclusion

Dans ce chapitre, nous avons détaillé notre approche de résumé vidéo, basée sur l'apprentissage profond en utilisant une architecture neuronale basée sur les réseaux de neurones récurrents à large « mémoire court-terme » (LSTM) qui prend les fonctionnalités spatiaux-temporelles présentes dans les images de la vidéo pour la génération dynamique du résumé.

CHAPITRE 5

PRÉSENTATION DES RÉSULTATS

Sommaire

5.1 Introduction 59
5.2 Environnement matériel 59
5.3 Environnement logiciel 59
5.3.1 Python 59
5.3.2 Numpy 60
5.3.3 OpenCV 60
5.3.4 TensorFlow 61
5.3.5 Keras 61
5.3.6 Le format HDF5 61
5.4 Ensemble de donnée (Dataset) 61
5.4.1 Office 62
5.4.2 Lobby 62
5.4.3 Campus 63
5.5 Outil de mesure 63
5.6 Résultat et discussion : 64
5.7 Conclusion 65

5.1 Introduction

Après avoir défini notre approche de création automatique de résumé vidéo ainsi que tous ces concepts liés, nous passons maintenant à l'expérimentation. Ce chapitre décrit l'environnement matériel et logiciel utilisé pour notre travail, ainsi que le jeu de test sur lequel on a travaillé et les mesures utilisées. Enfin nous terminerons ce chapitre par la présentation des résultats obtenus durant les tests.

5.2 Environnement matériel

Notre application va être réalisée sur une machine qui comporte les caractéristiques suivantes :

- **Marque :** Apple Macbook pro
- **Processeur :** Intel Core i7-4750HQ CPU @ 2.00GHz (Turbo Boost jusqu'à 3,2 GHz) avec 6 Mo de cache N3 partagé
- **Carte graphique :** Intel Iris Pro 1536 Mo
- **Mémoire :** 8 Go DDR3L à 1 600 MHz
- **Stockage :** 256 Go de stockage flash SSD
- **Système d'exploitation :** OS X 10.13.5

5.3 Environnement logiciel

5.3.1 Python

Python est un langage de programmation interprété (à ne pas confondre avec un langage compilé) créé par le néerlandais Guido Van Rossum au Centrum voor Wiskunde aux Pays-Bas en 2001. Le langage Python peut être exécuté directement sans passer par une phase de compilation. Il est possible de traduire un programme dans un langage (bytecode) qui est ensuite interprété par une machine virtuelle Python (mécanisme semblable au langage Java). Python est principalement inspiré du langage ABC (indentation comme syntaxe, ...), mais aussi du langage C et des outils Unix. Python est un langage libre placé sous licence PSFL (Python Software

Foundation License), il fonctionne sur de nombreuses plates-formes avec une grande communauté active. Python est aussi un langage orienté objet, il gère l'héritage de classe ainsi que l'héritage multiple (hérite de plusieurs classes) et le polymorphisme. [PACHON,]

Quelques avantages du language Python :

- proche du langage C.
- proche des langages fonctionnels.
- pas de perte de temps pour déclarer les types, variables, ...
- types de données complexes intégrés (listes, ...)
- permet d'intégrer d'autres codes cibles.
- possède un garbage collector (permettant de ne pas gérer les fuites de mémoire).

5.3.2 Numpy 

NumPy est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes. [Bressert, 2012]

5.3.3 OpenCV 

OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images et de vidéos en temps réel. Elle propose un ensemble de plus de 2500 algorithmes de vision par ordinateur ce qui permettre de donner à une machine le pouvoir d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plate-formes Windows, GNU/Linux, Android et MacOS. [Pulli *et al.*, 2012]

5.3.4 TensorFlow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache. Il est basé sur l'infrastructure DistBelief, initiée par Google en 2011, et est doté d'une interface Python. TensorFlow est l'un des outils les plus utilisés dans le domaine d'intelligence artificiel et l'apprentissage machine. [Abadi *et al.*, 2016]

5.3.5 Keras

Keras est une bibliothèque de réseau neurones open source écrite en Python. Il est capable de fonctionner sur TensorFlow, Microsoft Cognitive Toolkit, Theano ou MXNet.Conçu pour permettre une expérimentation rapide avec les réseaux neuronaux profonds, il se concentre sur la convivialité, la modularité et l'extensibilité. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). L'auteur et mainteneur principal de Keras est François Chollet, un ingénieur de Google. En 2017, l'équipe TensorFlow de Google a décidé de prendre en charge Keras dans la bibliothèque centrale de TensorFlow. Chollet a expliqué que Keras a été conçu pour être une interface plutôt qu'un cadre autonome d'apprentissage machine. Il offre un ensemble d'abstractions de plus haut niveau, plus intuitif, qui facilite le développement de modèles d'apprentissage en profondeur, quel que soit le backend informatique utilisé. [Chollet, 2017]

5.3.6 Le format HDF5

HDF5 vous permet de stocker d'énormes quantités de données numériques et de manipuler facilement les données de NumPy. Des milliers d'ensembles de données peuvent être stockés dans un seul fichier, catégorisés et étiquetés comme vous le souhaitez.

5.4 Ensemble de donnée (Dataset)

Dans ce qui suit, nous présentons les trois jeux de données vidéo multi-vues de notre dataset.

5.4.1 Office

Il s'agit du dataset Office fourni par [Fu *et al.*, 2010], qui a été pris avec 4 caméras stables mais non fixes dans un bureau (voir figure 5.1) . Les vibrations des caméras et le changement des conditions d'éclairage rendent difficile la production d'un bon résumé vidéo. Les quatre vidéos ne sont pas synchronisées, et certaines d'entre elles souffrent même d'une fréquence d'images instable.

FIGURE 5.1 – Une image d'exemple du dataset Office [Ou *et al.*, 2015]

5.4.2 Lobby

Il s'agit de l'ensemble de donnée Lobby fourni par [Fu *et al.*, 2010], qui a été pris avec 3 caméras dans un grand hall d'entrée. Cet ensemble de données a également été pris avec des caméras stables mais non fixes. Toutes les caméras sont synchronisées. Par rapport à l'ensemble de données de l'Office, cet ensemble de données contient plus de scènes surpeuplées avec des activités plus riches (voir figure 5.2), ce qui rend plus difficile à résumer.

FIGURE 5.2 – Une image d'exemple du dataset Lobby [Ou *et al.*, 2015]

5.4.3 Campus

Le dataset campus (Xu et al. 2016) contient des séquences vidéos de quatre scènes capturées par quatre caméras. différents d'autres dataset multi-vues dans la mesure où il met en point uniquement la tâche du suivi multi-objets. L'activités humaines est plus riches dans le dataset campus, avec chevauchement modéré dans les champs de vision entre les caméras [Qi *et al.*, 2017]

5.5 Outil de mesure

Afin de mesurer la qualité de notre résumé vidéo, on s'est intéressé à trois mesures souvent utilisées dans les travaux liés à l'apprentissage automatique, à savoir le recall, la précision et la F-measure. Afin de les calculer, on définit les valeurs dans le tableau suivant :

	Nombre de frames Pertinents	Nombre de frames Non Pertinents	Total
Nombre de frames Retrouvé	a	b	a+b
Nombres de frames non Retrouvé	c	d	c+d
Total	a+c	b+d	a+b+c+d

TABLE 5.1 – Calcul des paramètres Recall, Precision est F-mesure

Où :

- **Retrouvé :** signifie que les frames existent dans le résumé proposé.
- **Pertinent :** signifie que les frames existent dans le résumé généré (créé)

- **Rappel :** Le rappel mesure la capacité du système à restituer l'ensemble de frames pertinents. (Rappel exact par rapport à l'ensemble de frames retrouvées), obtenue en calculant [Baccini *et al.*, 2010] :

$$Rappel = \frac{Nombre\ de\ frames\ pertinents\ retrouve}{nombre\ de\ frames\ pertinents} = \frac{a}{a+c}$$

- **Précision :** Mesure la capacité du système à ne restituer que des frames pertinents [Deshpande, 2016], elle est définit comme suit :

$$Precision = \frac{Nombre\ de\ frames\ pertinents\ retrouve}{nombre\ de\ frames\ retrouvs} = \frac{a}{a+b}$$

- **F-Measure** Mesure qui combine le rappel et la précision. En effet, le rappel et la précision ont tendance à varier en sens inverse. [Baccini *et al.*, 2010], elle est définit comme suit :

$$F-Measure = 2\,.\,\frac{Rappel\ *\ Precision}{Rappel\ +\ Precision}$$

5.6 Résultat et discussion :

Méthodes	Office			Campus			Lobby		
	P	R	F	P	R	F	P	R	F
Random Walk	100	61	76,19	70	55	61,56	100	77	86,81
[Panda et al., 2016]	100	73	84,48	84	69	75,42	100	79	88,26
Notre approche	100	77	86,91	83	69	75,47	100	86	89,40

TABLE 5.2 – Comparaison de performance par rapport aux différentes approches(Précision, Rappel, F-measure)

Le tableau 5.2 montre les résultats de résumé vidéo sur les trois jeux de données multi-vues « Office, Campus et Lobby ». Pour les deux ensembles de jeu de données « Office et Lobby », notre approche produit des résumés avec la même précision que RandowWalk et [Panda *et al.*, 2016]

Cependant, l'amélioration de la valeur de rappel « Recall » environ de 16% pour Office et environ de 9% pour Lobby indique la capacité de notre méthode à conserver des informations plus importantes dans le résumé par rapport à RandomWalk. Amélioration de la valeur de de F-mesure environ 10% pour Office et environ de 3% pour Lobby par rapport à Random Walk et de 1% par rapport à [Panda *et al.*, 2016]. Dans l'ensemble, sur tous les ensembles de données, notre approche est supérieure à toutes les lignes en termes de F-measure.

5.7 Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et logiciel sur lesquels nous avons travaillé, ainsi que les différents résultats obtenus pour le jeu de données « office, lobby et campus ».

Conclusion Générale

Dans ce mémoire nous nous sommes intéressés au développement d'un outil efficace qui permet de gérer une base des fichiers vidéo. Cet outil consiste en un mécanisme qui résume le contenu vidéo multi-vue dans un réseau de caméras.

La réalisation d'un tel outil est d'une grande utilité; il permet d'extraire des informations utiles et récapitulatives, ce que nous fait gagner un temps considérable.

Notre travail consistait à prédire pour chaque image de la vidéo une probabilité pour indiquer si l'image est sélectionnée ou non dans le résumé final. Nous avons proposé une méthode pour la génération du résumé vidéo multi-vue qui se base sur l'apprentissage profond.en utilisant une architecture neuronales C3D afin d'extraire toutes les caractéristiques profondes de la vidéo. Ensuite nous avons utilisés une architecture neuronale basée sur les réseaux de neurones récurrents à longue « mémoire court-terme » (LSTM) qui prend les fonctionnalités spatiaux-temporelles présentes dans les images de la vidéo pour la génération dynamique du résumé final.

- **Perspectives :** Bien qu'on ait aboutit à de bons résultats, le travail peut être amélioré :
- L'utilisation d'un réseau génératif (Generative Adversarial Network),qui est bien adapté aux cas de la construction d'un résumé vidéo et qui produit de meilleurs résultats.
- Améliorer le temps d'extraction des caractéristiques C3D, qui malgré leur efficacité, reste cependant un processus long.

Bibliographie

[Abadi *et al.*, 2016] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow : Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.

[Abdelhamid, 2012] DJEFFAL Abdelhamid. *Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données.* PhD thesis, Université Mohamed Khider - Biskra, 2012.

[Baccini *et al.*, 2010] Alain Baccini, Sébastien Déjean, Nongdo Désiré Kompaoré, and Josiane Mothe. Analyse des critères d'évaluation des systèmes de recherche d'information. *Technique et Science Informatiques*, 29(3) :289–308, 2010.

[Baccouche, 2013] Moez Baccouche. *Neural learning of spatio-temporal features for automatic video sequence classification.* Theses, INSA de Lyon, July 2013.

[Bay *et al.*, 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3) :346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[Bengio, 2009] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1) :1–127, January 2009.

[Berglund *et al.*, 2015] Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha Karhunen. Bidirectional recurrent neural networks as generative models - reconstructing gaps in time series. *CoRR*, abs/1504.01575, 2015.

[Bilge Gunsel, 1997] A. Murat Tekalp Bilge Gunsel, Yue Fu. Hierarchical temporal video segmentation and content characterization, 1997.

[Boukadida, 2015] Haykel Boukadida. *Automatic video summarization using constraint satisfaction programming*. Theses, Université Rennes 1, December 2015.

[Bressert, 2012] E. Bressert. *SciPy and NumPy : An Overview for Developers*. O'Reilly Media, 2012.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[Chollet, 2017] Francois Chollet. *Deep Learning with Python*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2017.

[Cisco, 2016] Cisco. IP Video Surveillance Design Guide - Planning and Design. `https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/IPVS/IPVS_DG/IPVS-DesignGuide/IPVSchap4.html`, 2016.

[Collis, 2017] Jaron Collis. Glossary of Deep Learning : Autoencoder. `https://medium.com/deeper-learning/glossary-of-deep-learning-autoencoder-1044ec82c300`, 2017. [Online; accessed May 26, 2017].

[Comon, 1994] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36 :287–314, 1994.

[Crouspeyre, 2017] Charles Crouspeyre. Comment les Réseaux de neurones à convolution fonctionnent. `http://medium.com/@CharlesCrouspeyre/comment-les-réseaux-de-neurones-à-convolution-fonctionnent-b288519dbcf8`, 2017. [Online; accessed 17-July-2017].

[Dalal and Triggs, 2005] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *Interna-*

tional Conference on Computer Vision & Pattern Recognition (CVPR '05), volume 1, pages 886–893, San Diego, United States, June 2005. IEEE Computer Society.

[Deshpande, 2016] Adit Deshpande. Artificial Intelligence - Functional programming. `https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html`, 2016.

[Feuilloy, 2009] Mathieu Feuilloy. *Study of machine learning algorithms for syncope prediction.* Theses, Université d'Angers, July 2009.

[fouché, 2017] Edouard fouché. Neural based Outlier Discovery. `https://edouardfouche.com/Neural-based-Outlier-Discovery/`, 2017.

[Frans, 2016] Kevin Frans. Variational Autoencoders Explained. `http://kvfrans.com/variational-autoencoders-explained/`, 2016. [Online; accessed 06 August 2016].

[Fu *et al.*, 2010] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z. H. Zhou. Multi-view video summarization. *IEEE Transactions on Multimedia*, 12(7) :717–729, Nov 2010.

[Gelly, 2017] Gregory Gelly. *Speech processing using recurrent neural networks.* Theses, Université Paris-Saclay, September 2017.

[Girgensohn, 2003] A. Girgensohn. A fast layout algorithm for visual video summaries. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 2, pages II–77–80 vol.2, July 2003.

[Glorot, 2014] Xavier Glorot. *Apprentissage des réseaux de neurones profonds et applications en traitement automatique de la langue naturelle.* PhD thesis, Montréal : Thèse présentée à la Faculté des arts et des sciences en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.) en informatique, 2014.

[Goodfellow *et al.*, 2014] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[Guironnet, 2006] Mickael Guironnet. *METHODS OF VIDEO SUMMARIZATION FROM LOW LEVEL INFORMATION, CAMERA MOTION OR VISUAL ATTENTION.* Theses, Université Joseph-Fourier - Grenoble I, October 2006.

[Hawarah, 2008] Lamis Hawarah. *A Probabilistic Approach to Classify Incomplete Objects in a Decision Tree*. Theses, Université Joseph-Fourier - Grenoble I, October 2008.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and JÃŒrgen Schmidhuber. Long short-term memory. 9 :1735–80, 12 1997.

[Kingma and Welling, 2013] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[Kopf *et al.*, 2004] S. Kopf, T. Haenselmann, D. Farin, and W. Effelsberg. Automatic generation of video summaries for historical films. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 3, pages 2067–2070 Vol.3, June 2004.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[Larochelle, 2009] Hugo Larochelle. *Étude de techniques d'apprentissage non-supervisé pour l'amélioration de l'entraînement supervisé de modèles connexionnistes*. PhD thesis, University of Montréal, March 2009.

[Larsen *et al.*, 2015] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015.

[Lauly, 2016] Stanislas Lauly. *Exploration des réseaux de neurones à base d'autoencodeur dans le cadre de la modélisation des données textuelles*. PhD thesis, UNIVERSITÉ DE SHERBROOKE, Québec, Canada, 2016.

[LeCun *et al.*, 1990] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann, 1990.

[Liou *et al.*, 2008] Cheng-Yuan Liou, Jau-Chi Huang, and Wen-Chie Yang. Modeling word perception using the elman network. *Neurocomputing*, 71(16) :3150 – 3157, 2008. Advances in

Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006).

[Liou *et al.*, 2014] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139 :84 – 96, 2014.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, Nov 2004.

[Mahé, 2003] P. Mahé. Noyaux pour graphes et support vector machines pour le criblage virtuel de molécules. In *DEA MVA*, 2003.

[McCulloch and Pitts, 1943] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133, Dec 1943.

[Media, 2000] Adobe Dynamic Media. Initiation à la vidéo numérique, 2000.

[Microsoft, 2016] Philippe Beraud Microsoft. Quand le Deep Learning permet l'apprentissage par renforcement des robots industriels. `https://blogs.msdn.microsoft.com/mlfrance/2016/08/29/quand-le-deep-learning-permet-lapprentissage-par-renforcement-des-robots-industriels-1ere-partie/`, 2016.

[Mills *et al.*, 1992] Michael Mills, Jonathan Cohen, and Yin Yin Wong. A magnifier tool for video data. In *CHI*, 1992.

[Mioulet, 2015] Luc Mioulet. *Recurent neural network for handwriting recognition*. Theses, Université de rouen, July 2015.

[Mohamadally Hasan, 2006] Fomani Boris Mohamadally Hasan. Svm machine à vecteurs de support ou séparateur à vaste marge. In *BD Web, ISTY3,Versailles St Quentin*, 2006.

[Mohammed, 2014] KOUDRI Mohammed. Apprentissage automatique. In *UNIVERSITÉ ABOU BAKR BELKAID Tlemcen*, 2014.

[Money and Agius, 2008] Arthur G. Money and Harry Agius. Video summarisation : A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2) :121 – 143, 2008.

[Mostefaui Souad, 2015] Aichouch Hadjer Mostefaui Souad. Génération des résumés de vidéo, 2015.

[Mouelhi-Chibani, 2009] Wiem Mouelhi-Chibani. *Autonomous learning of neurone networks for the real-time monitoring and control of production based on optimization via simulation*. Theses, Université Blaise Pascal - Clermont-Ferrand II, October 2009.

[Nagasaka and Tanaka, 1991] Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances. In *VDB*, 1991.

[Nakache and Métais, 2005] Didier Nakache and Elisabeth Métais. Evaluation : nouvelle approche avec juges. In *Actes du XXIIIème Congrès INFORSID, Grenoble, France, 24-27 mai, 2005* [2005], pages 555–570.

[Nefy, 2017] Nefy. Deep Leafs Operational Classification Model. `http://nefy.org/2017/04/02/Deep-Leafs-Operational-Classification-Model/`, 2017.

[Nezhad *et al.*, 2018] Milad Zafar Nezhad, Dongxiao Zhu, Najibesadat Sadati, and Kai Yang. A predictive approach using deep feature learning for electronic medical records : A comparative study. *CoRR*, abs/1801.02961, 2018.

[Nikita Kaushik, 2016] Anshika Bhalla Nikita Kaushik, Ritu Rawat. A brief study of different feature detector and descriptor. Graphic Era University, Dehradun, India - International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 4, 2016.

[OpenAI, 2016] OpenAI. Generative Models. `https://blog.openai.com/generative-models`, 2016. [Online; accessed June 16, 2016].

[OpenDeep, 2015] OpenDeep. Tutorial : Your First Model (DAE) : Write your own denoising autoencoder and train it on MNIST. `http://www.opendeep.org/v0.0.5/docs/tutorial-your-first-model`, 2015.

[Ou *et al.*, 2015] S. H. Ou, C. H. Lee, V. S. Somayazulu, Y. K. Chen, and S. Y. Chien. On-line multi-view video summarization for wireless video sensor network. *IEEE Journal of Selected Topics in Signal Processing*, 9(1) :165–179, Feb 2015.

[P. Mahé, 2003] L. Ait-Ali P. Mahé. Projet d'apprentissage statistique svm pour l'apprentissage non supervisé. In *DEA MVA*, 2003.

[PACHON,] Cyril-Alexandre PACHON. Artificial Intelligence - Functional programming. `https://www.supinfo.com/cours/3AIT/chapitres/06-python`, -.

[Panda and Roy-Chowdhury, 2017] R. Panda and A. K. Roy-Chowdhury. Multi-view surveillance video summarization via joint embedding and sparse optimization. *IEEE Transactions on Multimedia*, 19(9) :2010–2021, Sept 2017.

[Panda *et al.*, 2016] Rameswar Panda, Abir Das, and Amit K. Roy-Chowdhury. Video summarization in a multi-view camera network. *CoRR*, abs/1608.00310, 2016.

[Peng and Ngo, 2005] Yuxin Peng and Chong-Wah Ngo. Hot event detection and summarization by graph modeling and matching. In Wee-Kheng Leow, Michael S. Lew, Tat-Seng Chua, Wei-Ying Ma, Lekha Chaisorn, and Erwin M. Bakker, editors, *Image and Video Retrieval*, pages 257–266, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Pibre *et al.*, 2016] Lionel Pibre, Marc Chaumont, Dino Ienco, and Jérôme Pasquet. Étude des réseaux de neurones sur la stéganalyse. In *CORESA : COmpression et REprésentation des Signaux Audiovisuels*, Nancy, France, May 2016.

[Porter *et al.*, 2001] Sarah Porter, Majid Mirmehdi, and Barry Thomas. Detection and classification of shot transitions. In *In Proc. of the 12th British Machine Vision Conf., pages 73 – 82*, 2001.

[Pulli *et al.*, 2012] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov. Real-time computer vision with opencv. *Commun. ACM*, 55(6) :61–69, June 2012.

[Qi *et al.*, 2017] Hang Qi, Yuanlu Xu, Tao Yuan, Tianfu Wu, and Song-Chun Zhu. Joint parsing of cross-view scenes with spatio-temporal semantic parse graphs. *CoRR*, abs/1709.05436, 2017.

[Rajendra, 2014] Sachan Priyamvada Rajendra. A survey of automatic video summarization techniques. 2014.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN : towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[Rosenblatt, 1958] F. Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[Rumelhart *et al.*, 1986] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing : Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[Saïda, 2011] BEDOUHENE Saïda. Recherche d'images par le contenu. UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU, DEPARTEMENT AUTOMATIQUE, Option : Traitement d'Images et Reconnaissance de Formes, 2011.

[science, 2017] Data science. What to do when data is missing. `http://curiousily.com/data-science/2017/02/02/what-to-do-when-data-is-missing-part-2.html`, 2017.

[Shibuya, 2017] Naoki Shibuya. Understanding Generative Adversarial Networks. `https://towardsdatascience.com/understanding-generative-adversarial-networks-4dafc963f2ef`, 2017.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[Smith and Kanade, 1995] Michael Smith and Takeo Kanade. Video skimming for quick browsing based on audio and image characterization. Technical Report CMU-CS-95-186, Carnegie Mellon University, Pittsburgh, PA, July 1995.

[Souad, 2014] TALEB ZOUGGAR Souad. *Contribution à l'apprentissage automatique symbolique par automates d'arbre et mesures de sélection.* Informatique & automatique, Université d'Oran, 2014.

[Sun and Kankanhalli, 2000] Xinding Sun and Mohan S. Kankanhalli. Video summarization using r-sequences. *Real-Time Imaging*, 6(6) :449 – 459, 2000.

[Sutton-Charani, 2014] Nicolas Sutton-Charani. *Learning from uncertain data and knowledge : application to the natural rubber quality prediction*. Theses, Université de Technologie de Compiègne, May 2014.

[Taniguchi *et al.*, 1995] Yukinobu Taniguchi, Akihito Akutsu, Yoshinobu Tonomura, and Hiroshi Hamada. An intuitive and efficient access interface to real-time incoming video based on automatic indexing. In *ACM Multimedia*, 1995.

[Thong, 2015] William Thong. Apprentissage de représentations pour la classification d'images biomédicales (mémoire de maîtrise, École polytechnique de montréal). tiré de https ://publications.polymtl.ca/1842/. 2015.

[Tonomura *et al.*, 1993] Yoshinobu Tonomura, Akihito Akutsu, Kiyotaka Otsuji, and Toru Sadakata. Videomap and videospaceicon : tools for anatomizing video content. In *INTERCHI*, 1993.

[Tran *et al.*, 2014] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D : generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[Uchihashi *et al.*, 1999] Shingo Uchihashi, Jonathan Foote, Andreas Girgensohn, and John Boreczky. Video manga : Generating semantically meaningful video summaries. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, MULTIMEDIA '99, pages 383–392, New York, NY, USA, 1999. ACM.

[Ueda *et al.*, 1991] Hirotada Ueda, Takafumi Miyatake, and Satoshi Yoshizawa. Impact : an interactive natural-motion-picture dedicated multimedia authoring system. In *CHI*, 1991.

[van der Maaten *et al.*, 2008] Laurens van der Maaten, Geoffrey E. Hinton, and Yoshua Bengio. Visualizing data using t-sne. 2008.

[Vidal and Vidal, 2006] José M Vidal and José M. Vidal. Fundamentals of multiagent systems, 2006.

[Wong, 2017] William Wong. BrainChip Enters AI Territory with Spiking Neural Network. `http://www.electronicdesign.com/embedded-revolution/brainchip-enters-ai-territory-spiking-neural-network`, 2017.

[Xiong *et al.*, 2006] Z. Xiong, R. Radhakrishnan, A. Divakaran, Z. Yong-Rui, and T.S. Huang. *A Unified Framework for Video Summarization, Browsing & Retrieval : with Applications to Consumer and Surveillance Video.* Elsevier Science, 2006.

[Yahiaoui, 2003] Bernard Yahiaoui, Itheri; Merialdo. *Construction automatique de résumés vidéos.* PhD thesis, 2003. Thèse de doctorat Signal et image Paris, ENST 2003.

[Zeiler and Fergus, 2013] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[Zhang *et al.*, 1995] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing : An integrated and content-based solution. In *Proceedings of the Third ACM International Conference on Multimedia*, MULTIMEDIA '95, pages 15–24, New York, NY, USA, 1995. ACM.

[Zhuang *et al.*, 1998] Yueting Zhuang, Yong Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, volume 1, pages 866–870 vol.1, Oct 1998.