REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

Pour l'obtention

D'un diplôme de master 2 en informatique

Option : Génie des Systèmes Informatiques

THÈME :

Résumé vidéo multi-sources

Réalisé par :

AHMED CHAOUCH Saida

Soutenue le : **devant :**

Mlle ZAHRA Fatma Zohra	Promoteur
Mr KAMECHE Abdallah Hichem	Encadreur
Mr CHÉRIF ZAHAR	Président
Mlle MEZZI MELYARA	Examinateur

2016/2017

Sommaire

Chapitre III : Apprentissage automatique

Chapitre IV: Approche proposé

Chapitre V: Présentation des résultats

Liste des tableaux :

Liste des figures :

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 des 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 et sur le principe de subspace clustring pour la sélection des trames clés « Key-Frames » dont les résultats s'avèrent satisfaisants.

Mots clés : *résumé vidéo, multi-sources, apprentissage profond, caractéristiques profondes*

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 work we propose a solution which consists in developing an application for multi-video summary generation. Our approach is based on deep learning for the extraction of characteristic vectors and on subspace clustering for the selection of key frames.

Key words : *vidéo-summarization, multi-view, deep learning, deep features, subspace clustering.*

نبذة مختصرة:

المراقبة بالفيديو هو نظام مراقبة يعتمد على كاميرات موضوعة في مساحات عمومية لإدارة المخاطر. للاستخدام الفعال لهذه الكاميرات، يجب على المستخدم أن ينظر إلى الصور ويجيب على أي نشاط مشبوه. مستخدمين مدربين و ذوي خبرة بإمكانهم القيام بالمتابعة بدقة. و لكن لعدد محدود من الفيديوهات. نظرا لتراكم السريع للفيديوهات المتواجدة على اجهزة الكمبيوتر مما توجب تطوير طرق للتحكم بيها و منها "خلاصة الفيديو".

معظم الابحاث تعطي الاهتمام عادة بخلاصة فيديو واحد، قليل من اهتم بمشكل خلاصة متعدد الفيديوهات و يأخذ بعين الاعتبار القيود و العناصر المطلوبة، نذكر على سبيل المثال إن المزيد من المعلومات موجودة بطريقة مماثلة في مختلف أشرطة الفيديو.

في هذا العمل، نقترح حل تطوير تطبيق لجيل من خلاصة الفيديو متعدد المصدر يعتمد على التعليم المعمق من اجل استخراج خصائص و ذلك بمبدأ كتل الفضاء الفرعي من اجل تحديد الاطر الرئيسية من اجل نتائج مرضية.

الكلمات المفتاحية: *ملخص فيديو، التعليم المعمق، خصائص معمقة*

INTRODUCTION GENERALE

Introduction générale :

Ces temps-ci, les caméras sont partout : les espaces publics sont surveillés par plusieurs caméras afin d'augmenter la sécurité, les propriétés privées sont protégées au moyen de caméras, et les magasins utilisent des caméras pour prévenir les vols. Plusieurs systèmes de vidéosurveillance sont installés tous les jours pour lutter contre la croissance du sentiment d'insécurité.

Pour utiliser efficacement des caméras de surveillance, 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 caméras simultanément, et seulement pour une période limitée de temps, car la concentration des opérateurs diminue avec le temps. Ils ne répondent alors pas rapidement à des évènements importants ou pourraient même les manquer complètement. Pour la plupart des applications - comme par exemple la détection d'évènements particuliers comme l'abandon d'un bagage dans un aéroport ou une gare -, regarder toutes les images des caméras est pratiquement impossible. Afin d'aider les opérateurs dans leurs travail, des « *résumes vidéos* » peuvent être utilisés.

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 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 bonne qualité.

Pour cela le mémoire est organisé comme suit :

- **Chapitre 1 :** 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 2 :** Ce chapitre contient une étude comparative des travaux proposés dans la littérature pour la génération des résumés vidéo.
- **Chapitre 3 :** Dans ce chapitre, nous nous intéresserons à présenter les différents types d'apprentissage automatique de façon plus particulière les réseaux de neurones.
- **Chapitre 4 :** Dans ce chapitre nous allons décrire la méthode que nous avons proposée pour la génération du résumé vidéo à partir de plusieurs vidéos
- **Chapitre 5 :** Ce chapitre contient la présentation des résultats obtenus.

Chapitre I

Résumé Vidéo

Concepts de base liés à la vidéo

1. Introduction

La vision par ordinateur (aussi appelée vision artificielle, vision numérique) est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle ≪voit ≫ lorsqu'on la connecte à une ou plusieurs cameras. [45].

Parmi l'une des taches de la vision par ordinateur on trouve le résumé vidéo.

La croissance rapide d'énorme flux de vidéos qui se trouvent sur les ordinateurs personnels et d'autres équipements (tels que les documents vidéo créés à partir d'enregistrements satellites, les enregistrements des appareils médicaux ou les caméras de surveillance) nécessite le développement de nombreux outils pour leur manipulation. La création automatique de résumés vidéo est un outil performant qui permet de résumer le contenu général de la vidéo et de ne présenter que les parties les plus pertinentes. [36], sous forme d'une « séquence audiovisuelle » ou « d'un ensemble d'images représentatives ».

Le résumé vidéo nous permet de faire une synthèse du contenu d'une vidéo après avoir fait une analyse de ses différents composants, et d'extraire des informations utiles et récapitulatives des sujets traités dans la vidéo [36].

Dans ce chapitre on va définir la notion du résumé vidéo ainsi de décrire la structure des différents composants et caractéristique d'une vidéo.

2. Résume statique et résumé dynamique :

Tandis qu'il existe diverses méthodes de création de résumés vidéos [36], la visualisation des résumés résultants se fait souvent selon deux approches: le résumé est soit représenté sous une forme «statique», soit «dynamique»:

- Un résumé visuel statique R^V [36] est construit sous forme d'un ensemble d'images représentatives du contenu visuel de la vidéo. Cette représentation peut nous permettre d'avoir un accès direct aux différentes parties du document vidéo original. 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.
- Un résumé visuel R^V peut aussi être présenté sous une forme dynamique [36]. Cette représentation 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.

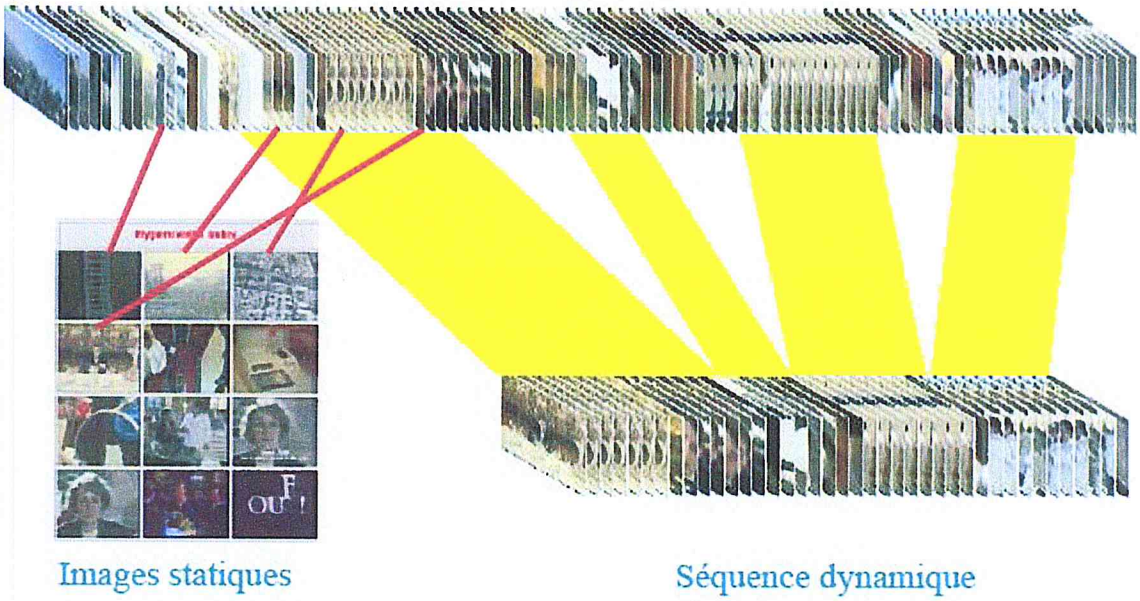


Fig 1 : représentation d'un résumé R^v [36]

3. Concepts élémentaires de la vidéo :

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 :

a. 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 de sons 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.[17] voir figure

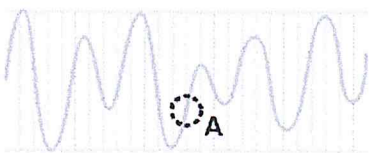


Fig 2 : représentation d'un signal analogique [17]

b. Signal numérique :

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

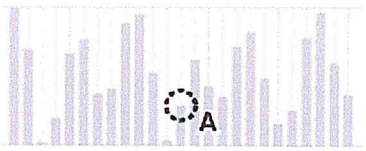


Fig 3 : représentation d'un signal numérique [17]

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.[17] voir figure

Fig 4 : représentation d'un signal numérique type binaire [17]

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.

c. Structure des vidéos :

Une vidéo se compose d'images affichées à une fréquence de 25 images (ou 30 images) par seconde mais en fait ce sont 50 trames par secondes (une image vidéo est constituée de deux trames), accompagnées d'une bande son. Suivant le regroupement des images, différentes entités peuvent être repérées.[42]

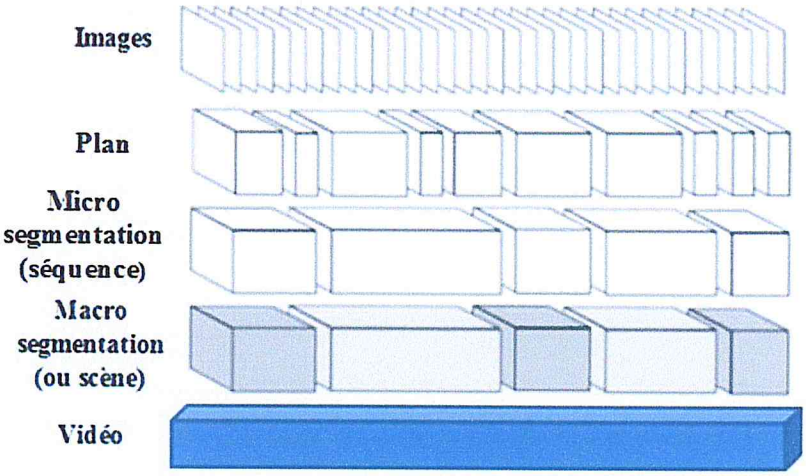


Fig 5 : structure d'une vidéo [42]

d. **Le plan** : souvent considéré 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 ont été proposés [42] :

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.

e. **La construction du plan:** le cadre met en valeur certains éléments, selon la composition de l'image (amorce de plan, premier plan, second plan, arrière-plan) [60]. La profondeur de champ dans la composition du plan est utilisée pour mettre en valeur les éléments que l'on souhaite comme la montre la figure suivante :

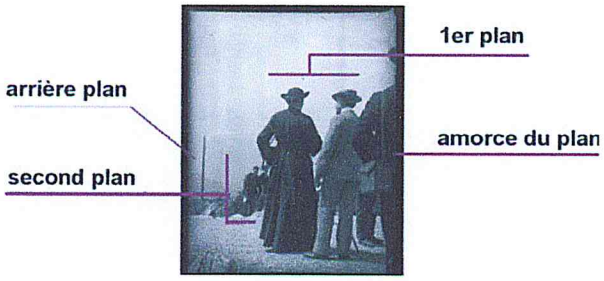


Fig 6 : construction du plan [68]

f. Coupe (shot) :

Elle est définie comme étant une transition immédiate d'une scène à l'autre qui se produit entre deux plans [32]. 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 [43]

Fig 7 : coupe par fondu enchainé [68]

   - soit par changement progressif de la couleur de la séquence jusqu'à atteindre une teinte uniforme (fade in/fade out) [52]

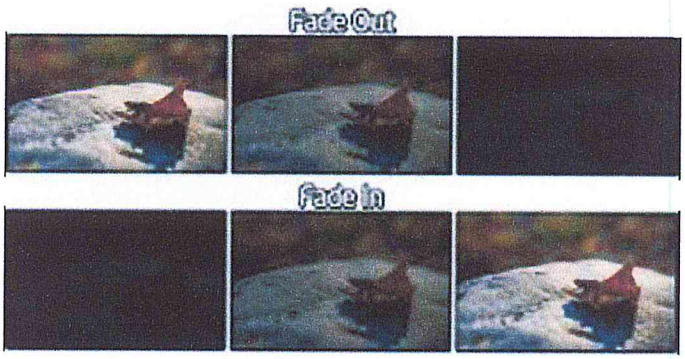


Fig 8 : coupe par changement progressif du couleur [68]

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 [32].

Fig 9 : coupe par changement brusque du plan [68]

g. Image numérique :

C'est un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain. Elle peut être décrite sous la forme d'une fonction I(x,y) de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur.[1]

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits).

L'image numérique (image matricielle) est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter [2]

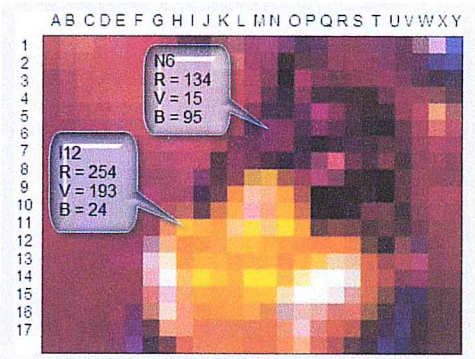


Fig 10 :éléments d'une image numérique

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy) en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques f(x,y) où :

x, y : coordonnées cartésiennes d'un point de l'image.

f(x, y) : niveau de gris en ce point

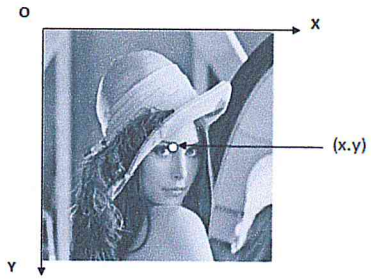


Fig 11 : coordonnées cartésiennes d'un point de l'image

➢ ***Pixel :*** Un pixel représente le plus petit élément d'une surface d'affichage, par exemple sur un écran d'ordinateur. On associe au pixel une couleur, et une intensité. Un pixel est lui-même composé de trois points de couleurs différentes : Rouge, Vert, Bleu. Ce sont comme des petites lampes capables de composer jusqu'à 16 millions de couleurs en jouant sur l'intensité du Rouge, du Vert ou du Bleu. C'est le principe de la synthèse additive des couleurs, qui concerne la lumière (écrans, vidéoprojecteurs, lampes).

Fig 12 : représentation d'un pixel

➢ ***La Définition*** d'une image matricielle correspond donc au produit du nombre de pixels qui compose l'image en Longueur (axe horizontal) par celui de sa Hauteur (axe vertical).

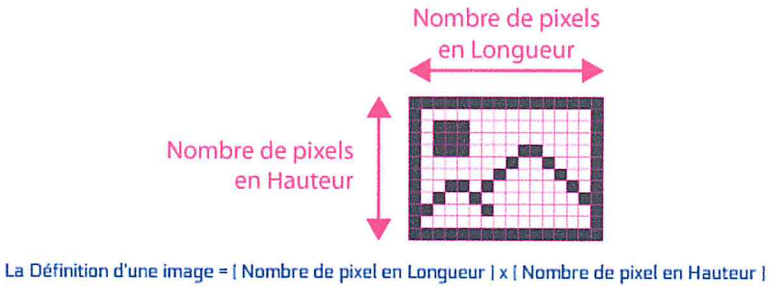


Fig 13(1) : définition d'une image

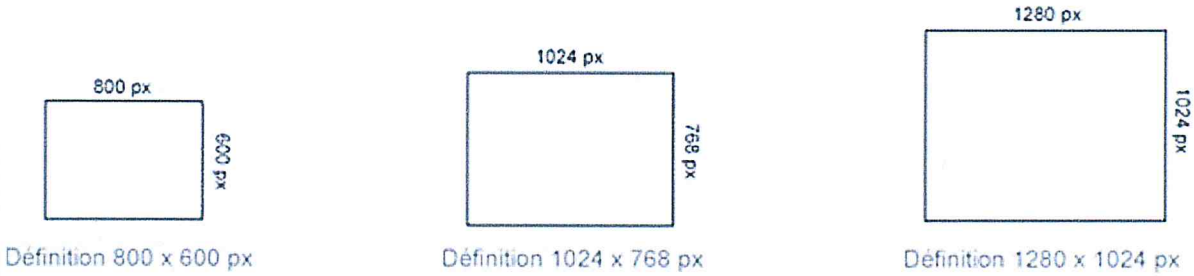


Fig 13(2) : définition d'une image

➢ **Résolution :** La résolution d'une image est définie par le nombre de pixels par unité de longueur. Usuellement, le nombre de pixels est compté par pouce (1 pouce = 2,54 cm, noté ppp ou dpi) ou par centimètre. Plus le nombre de pixels par unité de longueur est élevé, plus la quantité d'information décrivant l'objet est importante donc la résolution est grande. Ce paramètre est défini souvent lors de l'acquisition de l'image (réglage de l'appareil photo, résolution du logiciel du scanner.. . etc.) ou ultérieurement dans les logiciels de traitement d'images. La publication d'image sur Internet correspond souvent à une résolution de 90 ppp (points par pouce : un pouce = 25,4 mm) et dans la presse écrite de 150 ppp [40].

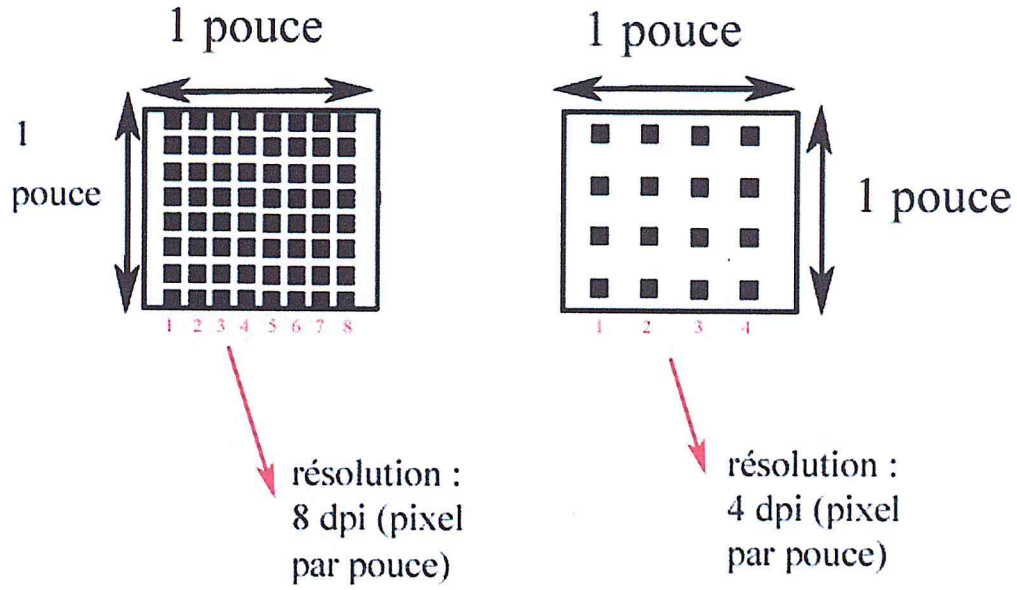


Fig 14 : résolution d'une image

En pratique, si vous avez une image de 4000x3000 pixels à une résolution de 300 dpi, elle aura une taille réelle (en gros) de 25x30 cm. Si vous baissez sa résolution à 200 dpi, elle aura une taille de 40x50 cm. Les pixels de l'image étant moins compressés, l'image prendra plus de place visuellement tout en perdant en qualité de détail. Sa définition est toujours la même (4000x3000) mais sa résolution a baissé. Si l'on met moins de pixels par pouce carré (dpi), l'image finale pourra être plus grande mais restituera moins de détails.

La résolution de 300 dpi a été choisie pour que la vision d'une image soit toujours bonne à notre œil pour les documents imprimables. Nous précisons que cette résolution concerne les images imprimables car les résolutions des écrans sont de 72 dpi.

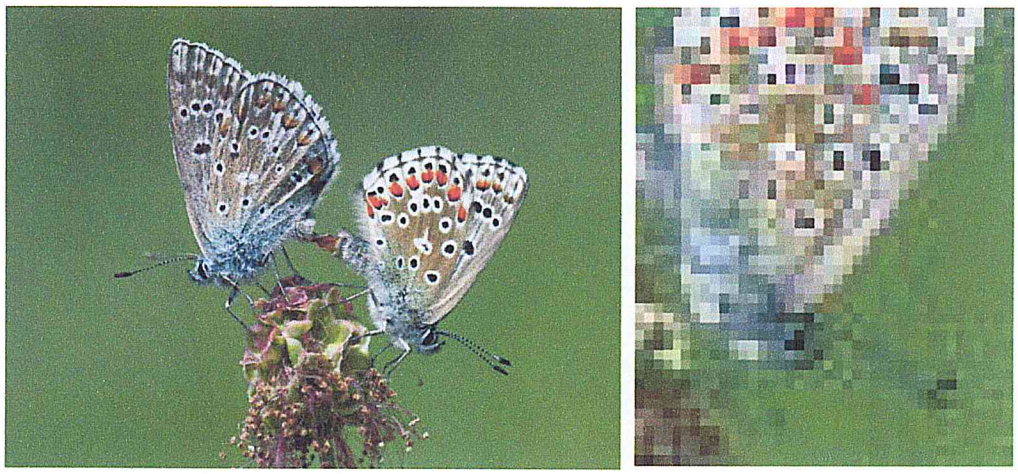


Fig 15 : images avec résolution différente

➢ ***Histogramme de couleur*** **:** L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image.

Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée. Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

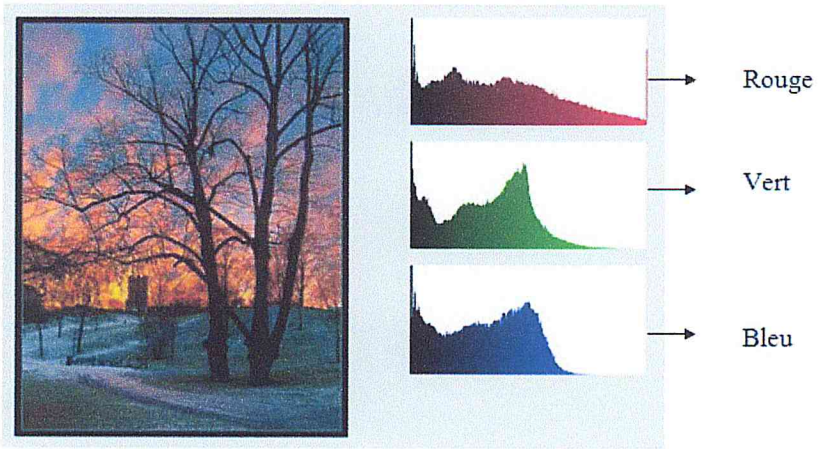


Fig 16 : histogramme de couleur

4- Détecteurs des points d'intérêt et 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.

4-1 Définition d'un point clé :

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. [65]

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.[65]

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

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 :

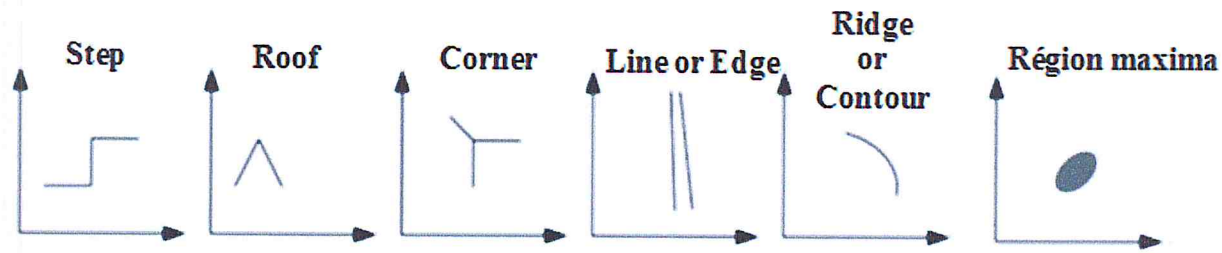


Fig 17 : Types de points clés, (De gauche à droite) Étape, toit, coin, ligne ou bord, arête ou contour, région maxima [65]

4-2 Scale Invariant Feature Transformation(SIFT) :

C'est un descripteur qui a été proposé par Lowe et donne des caractéristiques qui sont invariantes à la distorsion affine, l'échelle, les changements d'éclairage (illumination), le bruit, la rotation et les changements de point de vue 3D. L'algorithme SIFT comporte 4 étapes principales: [65]

- Détection de l'espace d'échelle
- Localisation des points clé

- Affectation d'orientation
- Génération des descriptions

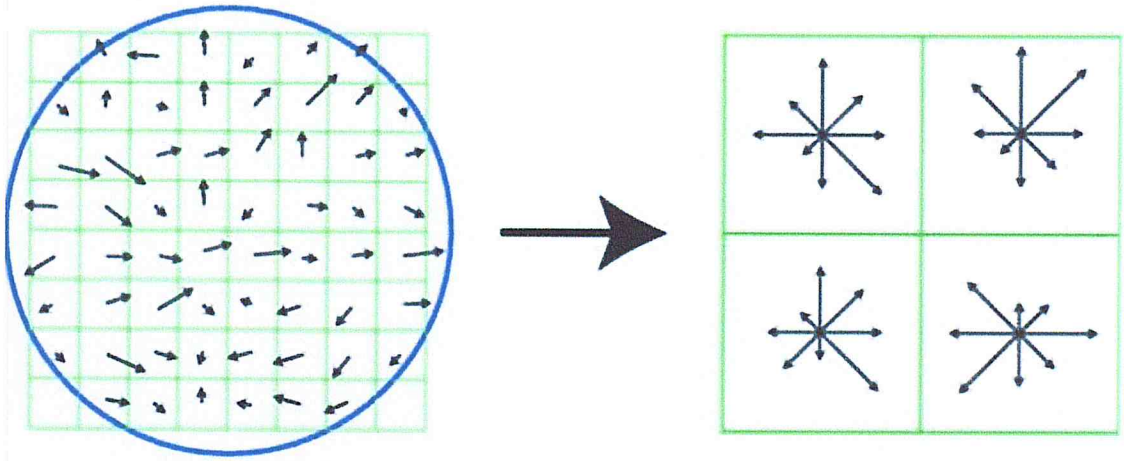


Image gradients **Keypoint descriptor**

Fig 18 : Descripteur SIFT [65]

La première étape consiste à reconnaître les échelles et l'emplacement par les extrêmes de l'espace d'échelle dans la fonction Différence-de Gaussienne (DoG) avec des valeurs distinctes de σ.

$$g_1(x,y) - g_2(x,y) = G_{\sigma_1} * f(x,y) - G_{\sigma_2} * f(x,y) = (G_{\sigma_1} - G_{\sigma_2}) * f(x,y) = DoG * f(x,y)$$

Où, f est l'image et G est la fonction de Gaussian. Pour générer une soustraction de DoG des images gaussiennes sont effectuées. Le sous-échantillon des images gaussiennes est effectué par un facteur 2, après quoi un DoG est généré par des images échantillonnées.

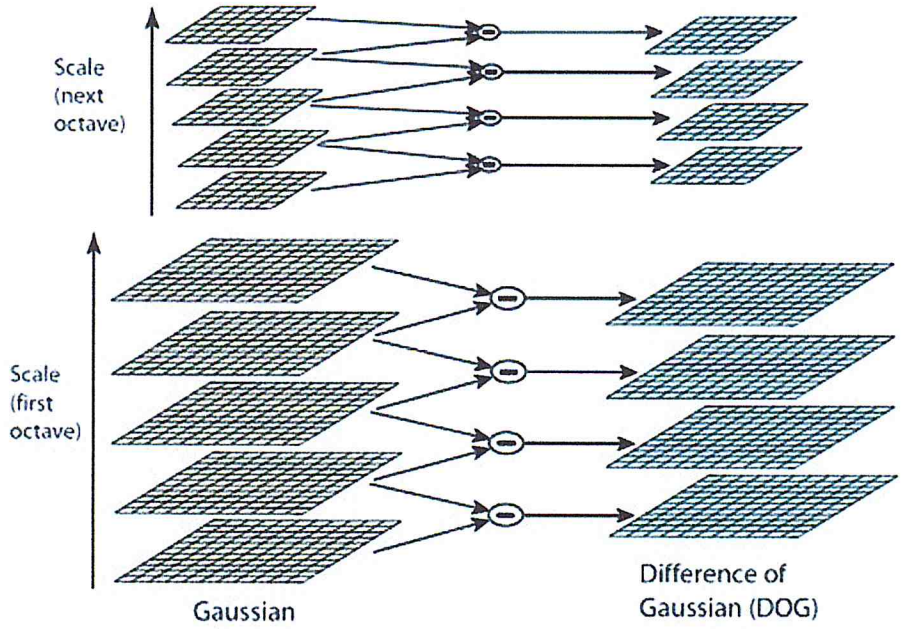


Fig 19 : DoG pyramid [65]

Pour détecter les minima et maxima locaux de D (x, y, σ), un pixel est comparé par voisinage 3 * 3. La figure Fig19 montre la pyramide DoG. Dans l'étape de localisation, les points clés sont localisés et filtrés en supprimant les points clés où les points de contraste faible sont rejetés par eux.

Dans l'étape d'affectation d'orientation, l'orientation du point clé d'obtention dépend du gradient d'image local. Dans l'étape de génération de description, le descripteur d'image local de chaque point clé dépend de l'orientation de l'image et de l'amplitude du gradient à chaque point d'échantillonnage dans un domaine centré au point clé [50,55].

Fig 20: Original image [65]

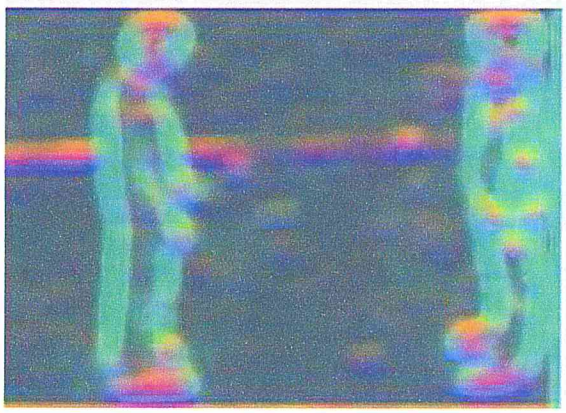
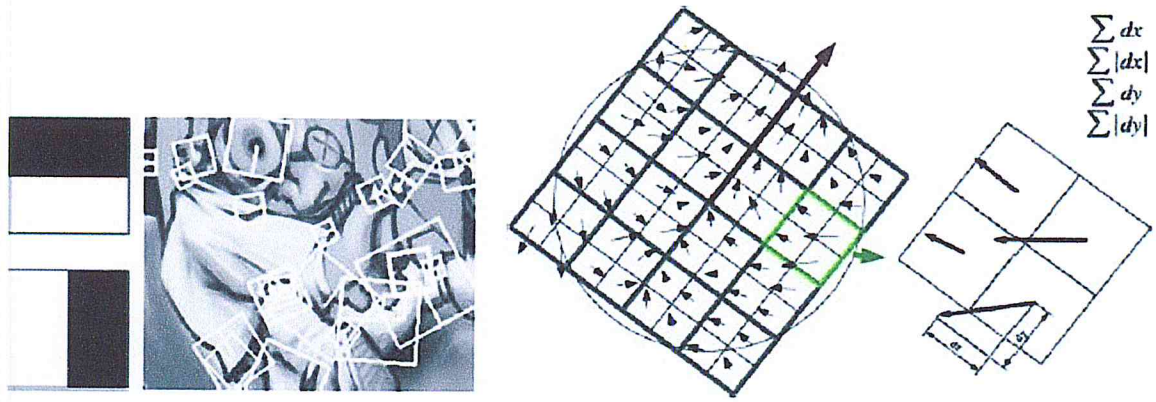


Fig 21: After apply SIFT descriptor [65]

4-3 Speeded-Up Robust Feature (SURF):

C'est un descripteur de fonctionnalité local et un détecteur. Il est également utilisé pour la reconnaissance d'objets, l'enregistrement, la reconstruction 3D et la classification. Il est inspiré du descripteur SIFT. Il (version standard) est plus rapide que SIFT. Il utilise l'ondelette Haar. SURF est meilleur en termes de vitesse.[65]



(a) Computing orientation of a patch around a keypoint (b) SURF descriptor

Fig 22 : Descripteur SURF [65]

➢ **Localisation de points clés:** la détection de structures de type Blob est effectuée lorsque l'élément de la Matrice Hessienne est maximal. Pour localiser le point d'intérêt Blob-like, il utilise un espace d'échelle sur un voisinage 3 * 3 * 3. Pour identifier l'orientation des caractéristiques, on calcule une série de réponses de type HAAR en domaine local entourant chaque point d'intérêt dans un rayon circulaire, calculé à l'échelle pyramidale correspondante pour le point d'intérêt. Le point donné x = (x, y) dans l'image I, la matrice Hessienne H (X, σ) dans X à l'échelle σ, on peut l'exprimer comme :[65]

$$\mathbf{H(X,\sigma)} = \begin{bmatrix} \mathbf{L_{xx}(x,\sigma)} & \mathbf{L_{xy}(x,\sigma)} \\ \mathbf{L_{yx}(x,\sigma)} & \mathbf{L_{yy}(x,\sigma)} \end{bmatrix}$$

Où Lxx (x, σ) est le résultat de convolution et même pour Lxy (x, σ) et Lyy (x, σ).

➢ **Point clé Description:** Sur les images intégrales, les acquittements d'ondelettes Haar sont appliqués dans les directions X et Y. Ensuite, l'orientation doit être régénérée en fonction des informations du domaine circulaire sur le point clé. Le

descripteur des points clés est calculé par une région carrée ajustée à l'orientation sélectionnée. À la fin, les caractéristiques sont adaptées [44, 61].

Fig 23: Original image [65]

Fig 24: After apply SURF descriptor [65]

4-4 Histogramme de gradient (HOG) :

Ce descripteur est utilisé dans le traitement des images et la vision par ordinateur pour la détection des objets. Dans les aires localisées d'images cette technique compte des apparences d'orientation de gradient [65]. Il est apparenté au descripteur de SIFT, à l'histogramme d'orientation de bord et à la forme, mais il est différent en termes de grille dense calculée de cellules uniformément espacées. HOG fonctionne sur les données non formées; Tandis que diverses méthodes dépendent du lissage gaussien et d'une autre technique de filtrage pour construire les données, HOG est créé en particulier pour utiliser toutes les données non formées sans insérer d'artefacts de filtrage qui suppriment des détails. C'est un compromis: le filtrage des artefacts comme le lissage par le rapport aux artefacts d'image tels que des détails raffinés. La méthode HOG représente les résultats préférentiels pour les données brutes.[65]

HOG taxonomie:

- Spectres: histogrammes de gradient de la région locale
- Forme de l'entité: Cercle ou rectangle
- Caractéristique: Densité de 64x128 rectangle typique
- Méthode de recherche: Grille sur l'espace de l'échelle
- Fonction de distance: Euclidienne
- Densité de caractéristiques: blocs denses et chevauchants
- Robustesse: 4 (bruit, illumination, échelle, point de vue)

Fig 25: Original image [65]

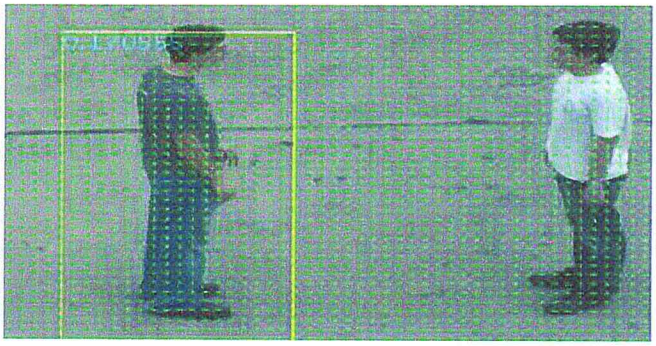


Fig 26: After apply HOG descriptor [65]

4-5 Binaire Robuste Invariant Scalable Points Clés (BRISK) :

Il s'agit d'une approche binaire locale. BRISK utilise une forme de région de motif symétrique circulaire et un agrégat de 60 paires de points comme segments de ligne disposés en 4 anneaux concentriques [47].

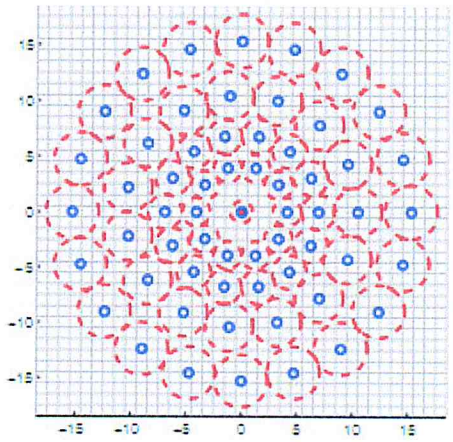


Fig 27: Descripteur BRISK [65]

BRISK prend des paires ponctuelles de segments longs et de segments courts, ce qui supporte une mesure de l'invariance de l'échelle, car la résolution grossière peut être mieux cartographiée par de longs segments et la résolution fine peut être cartographiée par des segments courts [66]. L'algorithme de BRISK est unique. Les principales étapes de calcul de l'algorithme sont données ci-dessous:

- La sélection des points clés dans l'espace d'échelle à l'aide d'AGHAST ou FAST.

- Pour obtenir la valeur de point, appliquez un lissage gaussien à chaque point d'échantillonnage de pixel.
- Donne 3 ensembles de paires: Paires courtes, paires longues et paires non utilisées (les paires non utilisées ne sont pas dans la paire courte ou la paire longue).
- Calculez les gradients entre les paires longues, les sommé pour guider l'orientation.
- L'orientation du gradient permet de faire pivoter et d'ajuster les paires courtes.
- Génère un descripteur binaire à partir de comparaisons ponctuelles à paires courtes.

Conclusion :

Dans ce chapitre, nous avons présenté quelques notions de base sur la vidéo, suivi d'une brève étude sur les différents descripteurs qui sont nécessaires pour décrire les caractéristiques visuelles essentielles (forme, couleur...) des objets et les caractéristiques de leur mouvements (trajectoires). Ses caractéristiques sont représentées sous formes des vecteurs que nous allons les détailler dans les chapitres suivants.

Chapitre II

Méthodes pour la génération

Des résumés vidéo

Introduction :

Différents travaux de recherche ont traité le problème de construction des résumés vidéos en apportant diverses solutions et propositions, malgré que ce soit un domaine de recherche assez récent mais en plein essor.

La plupart des systèmes cités dans la littérature utilisent uniquement le flux vidéo en faisant l'extraction des images ou des segments représentatifs. Les images représentatives (images clés) d'une vidéo donnée peuvent être sélectionnées à des intervalles de temps uniformes.
On distingue deux type de travaux : ceux exploitant une seule source vidéo, et ceux utilisant plusieurs sources vidéo.

I- Travaux basé sur une seule source vidéo:

Diverses méthodes ont été proposées. Ces dernières peuvent être groupées en différentes catégories :

- les méthodes basées sur un découpage en plans.
- les méthodes basées sur une classification
- ainsi que des méthodes basées sur d'autres alternatives.

1- Approches basées sur un découpage en plans :

Beaucoup de travaux [34] se sont concentrés sur le découpage de la vidéo en un ensemble de plans, et la recherche ultérieure d'un nombre d'images représentatives du contenu de chaque plan détecté.

Une fois le découpage de la vidéo en plans est effectué, plusieurs suggestions ont été faites pour la sélection d'une image caractéristique du plan : la première image [20], la dernière image, l'image médiane, l'image la plus proche de l'image moyenne du plan, la première image du plan [21], les deux images les plus différentes, etc…Lienhart et al.[22] et Ueda et al. [8] ont représenté chaque plan par ses première et dernière images. Ferman et al. [24] ont établis une classification des images de chaque plan. L'image la plus proche du centre de la plus grande classe a été sélectionnée comme étant l'image représentative du plan.

2- Approches basées sur une classification :

Plusieurs chercheurs [36] ont vu que l'emploi d'une phase de découpage en plans permet de diminuer la quantité d'information considérée dans la phase de sélection des images représentatives de la vidéo. Si différents plans, qui se retrouvent à des emplacements espacés dans la vidéo, ont des contenus très similaires alors les images sélectionnées de ces derniers seront certainement assez semblables. Ce qui provoquera une redondance d'information dans le résumé de la vidéo contenant ces images représentatives.

Dans le but de choisir des images représentatives qui sont différentes les unes des autres et qui représentent bien le contenu de la vidéo, [36] a vu qu'il est intéressant de comparer toutes les images de la vidéo entre elles. Quelques chercheurs ont proposé de faire une classification globale de l'ensemble des images [6, 26, 27], puis de sélectionner une image par classe afin d'être insérée dans le résumé comme étant une image représentative du contenu visuel de cette classe.

3- Diverses alternatives :

- Chiu et al. [29] ont proposé un algorithme génétique pour la segmentation des vidéos basée sur une fonction de similarité d'images contiguës [36]. Un chromosome est une chaîne de 0 et de 1 où chaque 0 correspond à une image de la vidéo et les 1 correspondent aux images considérées comme des limites des segments. La fonction de sélection est basée sur le calcul de la similarité des images.

- Stefanidis et al. [30] ont proposé une approche de construction d'un résumé vidéo basée sur l'analyse des trajectoires des objets appartenant à cette vidéo [36].

II- Travaux basé sur plusieurs sources vidéo (vidéo multi-sources):

Malgré que la construction d'un résumé d'une vidéo simple reçoive une attention croissante, il n'y a pas beaucoup de travaux consacrés au problème de construction de résumé multi–sources.

Les études précédentes de résumé vidéo se sont concentrées sur les résumés d'une seule vidéo, et les résultats ne seraient pas bons si elles étaient appliquées directement aux

vidéos multi-sources, en raison de problèmes tels que la redondance d'information dans plusieurs vues de la même scène.

Nous allons présenter quelques travaux faits sur l'extraction du résumé vidéo multi-sources.

1- Approche basé sur la théorie des graphes :

Pour Yanwei.al [49] : leur travail consiste à construire un graphe des shots spatio-temporel et de formuler le problème du résumé en tant que tâche d'étiquetage du graphique. Une telle représentation donne la possibilité de résoudre le problème de résumé multi-vues en utilisant la théorie des graphes.

- 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.[49]
- 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. [49]
- Le storyboard vidéo multi-vues et le forum événementiel sont présentés pour représenter le résumé vidéo multi-vu. 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. [49]

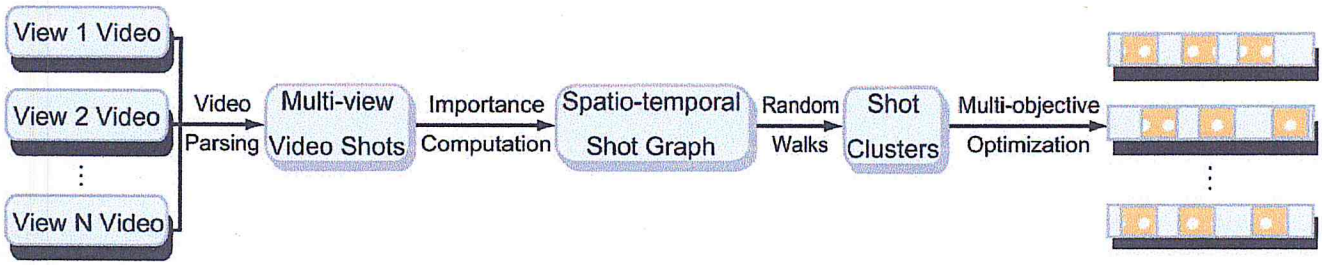


Fig 47 : Une vue d'ensemble du résumé vidéo multi-vue d'après Yanwei.al [49]

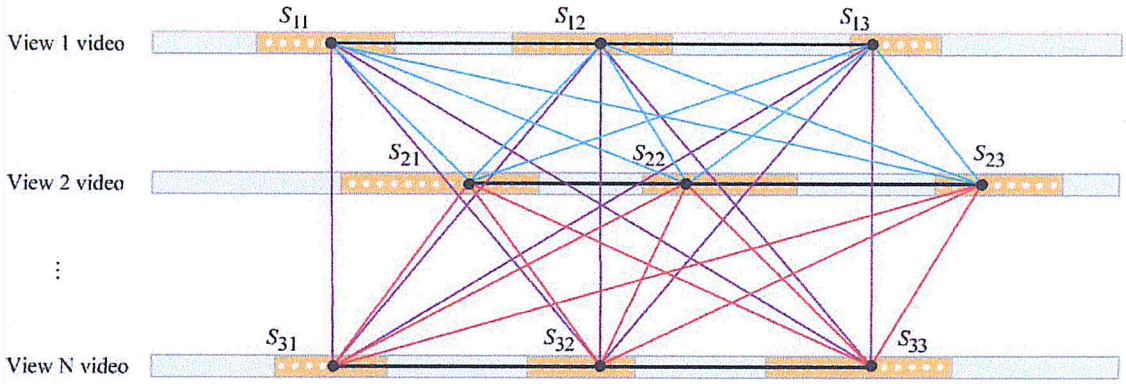


Fig 48: Spatio-temporal shot graph.[49]

2- Approche basé sur la classification par image-clé et la théorie de l'ensemble :

Ping Li, Yanwen Guo, Hanqiu Sun [51] : Ont proposé une carte de corrélation à image clé pour représenter naturellement les corrélations entre les multi-keyframes.

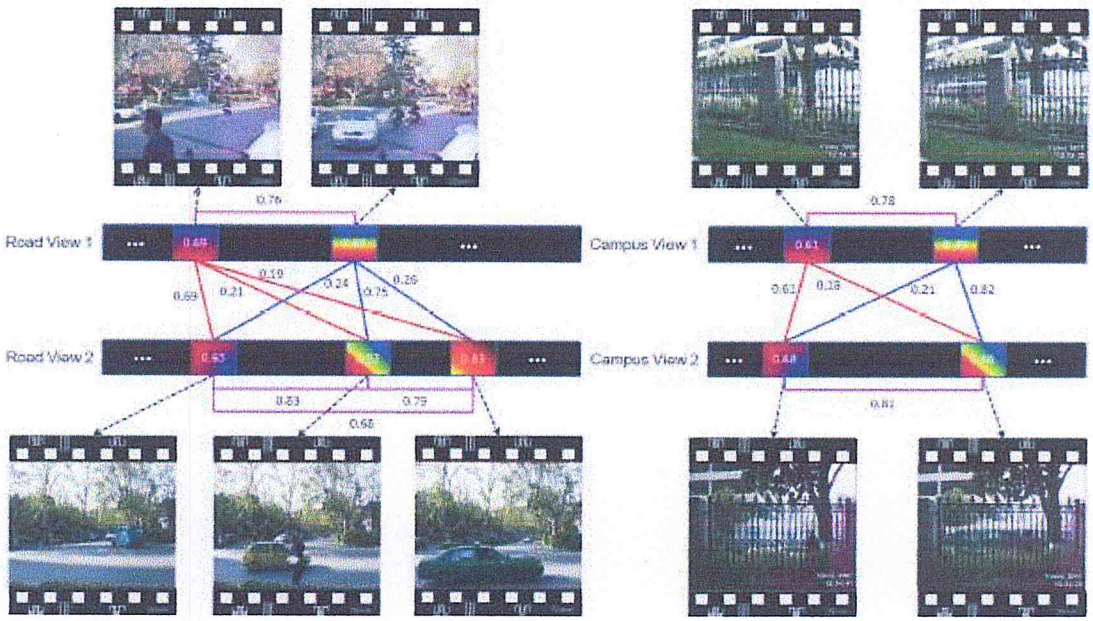


Fig 49: Event-centered correlation maps for the multi-keyframe abstractions [51]

Les corrélations pondérées sont calculées en fonction de la théorie probabiliste et de la similarité sémantique temporelle et visuelle entre les images clés. Ensuite ils ont classé la carte de corrélation via SVM et ont identifié les classes d'images clés centrées sur l'événement avec des corrélations internes via des pondération élevées. Les images clés essentielles sont également générées pour l'abstraction à l'aide d'un ensemble (Rough Sets), et les cartes de corrélation centrées sur l'événement sont présentées pour assembler en série plusieurs images-clés à travers le temps, afin de faciliter la navigation sur les jeux de données vidéo. [51]

Conclusion :

Dans ce chapitre, nous avons présenté une revue générale des approches de construction de résumés vidéo qui reposent sur des caractéristiques différentes. Le but de ce chapitre est de donner une vue d'ensemble de ce qui se fait dans le monde scientifique qui s'intéresse à la génération de résumés vidéo. A travers cette étude, nous avons aussi remarqué qu'il y a peu de travaux consacrés au cas des résumés multi-vidéos.

Nous avons observé que l'extraction des résumés vidéo est un thème de recherche très vaste qui nécessite la mise en œuvre de nouveaux outils plus performants qui répondent au mieux aux exigences des sciences et technologies nouvelles comme les réseaux de neurones qui et le but de notre étude.

Chapitre III

Apprentissage automatique

Introduction :

L'apprentissage automatique (machine learning en anglais), un des champs d'étude de l'intelligence artificielle, est la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.[70]

Il existe actuellement une multitude d'algorithmes d'apprentissage qui sont généralement catégorisés par rapport au mode d'apprentissage adopté. Ainsi, face à un problème particulier, il est nécessaire de choisir la forme d'apprentissage adaptée aux types de données que l'on doit traiter. Bien entendu, un même problème peut être abordé sous plusieurs angles et, par conséquent, plus d'une méthode d'apprentissage peut s'avérer pertinente à sa résolution. Les principales formes d'apprentissage automatiques sont : l'apprentissage supervisé, l'apprentissage non-supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement. Nous allons les détailler dans ce qui suit.

L'apprentissage profond est un ensemble de méthodes d'apprentissage automatique, 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, du traitement automatisé du langage et de la vision par ordinateur.[46].

Parmi l'une des taches de la vision par ordinateur on trouve le résumé vidéo.

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

1- Différents type (catégorie) d'apprentissage automatique :

Il y' a 4 différents types pour l'apprentissage automatique : apprentissage non supervisé, apprentissage semi-supervisé, apprentissage supervisé et enfin apprentissage par renforcement, on va les détailler dans ce qui suit.

1-1 Apprentissage non supervisé :

La classification non supervisée - ou clustering - cherche à construire une partition d'un jeu de données de telle sorte que les données au sein d'un même groupe exhibent des propriétés ou des caractéristiques communes et qui les distinguent des données

contenues dans les autres groupes. A ce titre, les méthodes de clustering ont été largement utilisées dans de nombreux domaines d'applications allant de la biologie (classification de protéines ou de séquences de génomes), à l'analyse de documents (textes, images, vidéos).[53]

1-2 Apprentissage supervisé : (supervised learning)

La classification supervisée, dite aussi discrimination est la tâche qui consiste à discriminer des données, de façon supervisée (ç-à-d avec l'aide préalable d'un expert), un ensemble d'objets ou plus largement de données, de telle manière que les objets d'un même groupe (appelé classes) sont plus proches (au sens d'un critère de similarité choisi) les unes aux autres que celles des autres groupes. Généralement, on passe par une première étape dite d'apprentissage où il s'agit d'apprendre une règle de classification à partir de données annotées (étiquetées) par l'expert et donc pour lesquelles les classes sont connues, pour prédire les classes de nouvelles données, pour lesquelles les données sont inconnues [56]. La prédiction est une tâche principale utilisée dans de nombreux domaines, y compris l'apprentissage automatique, la recherche d'information, la reconnaissance de formes, le traitement de signal et d'images.

Les données traitées en classification peuvent être des images, signaux, textes, autres types de mesures, etc. Dans le cadre de cette thèse les données seront des données visuelles (images).

1-2-1 Définition d'une classe :

Une classe (ou groupe) est un ensemble de données formée par des données homogènes (qui se ressemblent au sens d'un critère de similarité « distance, densité de probabilité, etc »). Par exemple, une classe peut être une région dans une image.[56]

Pour notre étude du résumé vidéo, on va s'intéresser à cette catégorie d'apprentissage pour déterminer si une image est une keyframe ou non et aussi pour la détection d'objet dans une image. Plusieurs modèles d'apprentissage supervisé de caractéristiques ont ainsi été proposés dans la littérature. Nous aborderons par la suite quelques uns.

1-3 Apprentissage semi-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 [63].

1-4 Apprentissage par renforcement :

L'apprentissage par renforcement concerne l'apprentissage d'actions à effectuer dans un environnement changeant afin de maximiser une récompense totale. [58].

Pour ce travail on se focalise sur l'apprentissage supervisé qui est plus adapté à notre étude, et dans ce qui suit on aborde les techniques les plus utilisées.

2- Quelques algorithmes (modèles) d'apprentissage supervisé :

2-1 Arbres de Décision :

Les arbres de décision sont composés d'une structure hiérarchique en forme d'arbre. Cette structure est construite grâce à des méthodes d'apprentissage par induction à partir d'exemples. L'arbre ainsi obtenu représente une fonction qui fait la classification d'exemples, en s'appuyant sur les connaissances induites à partir d'une base d'apprentissage. En raison de cela, ils sont aussi appelés arbres d'induction (Induction Decision Trees). Une définition un peu plus formelle des arbres de décision est la suivante : un arbre de décision est un graphe orienté, sans cycles, dont les noeuds portent une question, les arcs des réponses, et les feuilles des conclusions, ou des classes terminales [39].

Un arbre de décision se construit à partir d'un ensemble d'apprentissage. Un ensemble de questions sur les attributs est construit afin de partitionner l'ensemble d'apprentissage en sous-ensembles qui deviennent de plus en plus petits jusqu'à ne contenir à la fin que des observations relatives à une seule classe. Les résultats des tests forment les branches de l'arbre et chaque sous-ensemble en forme les feuilles. Le classement d'un nouvel exemple se fait en parcourant un chemin qui part de la racine pour aboutir à une feuille. La Figure.28 donne un exemple d'arbre de décision pour le classement d'un ensemble de cas, avec un test d'appartenance à une classe. Dans ce cas particulier, les cas dits positifs sont ceux qui appartiennent à la classe et les cas dits négatifs sont ceux qui n'y appartiennent pas.

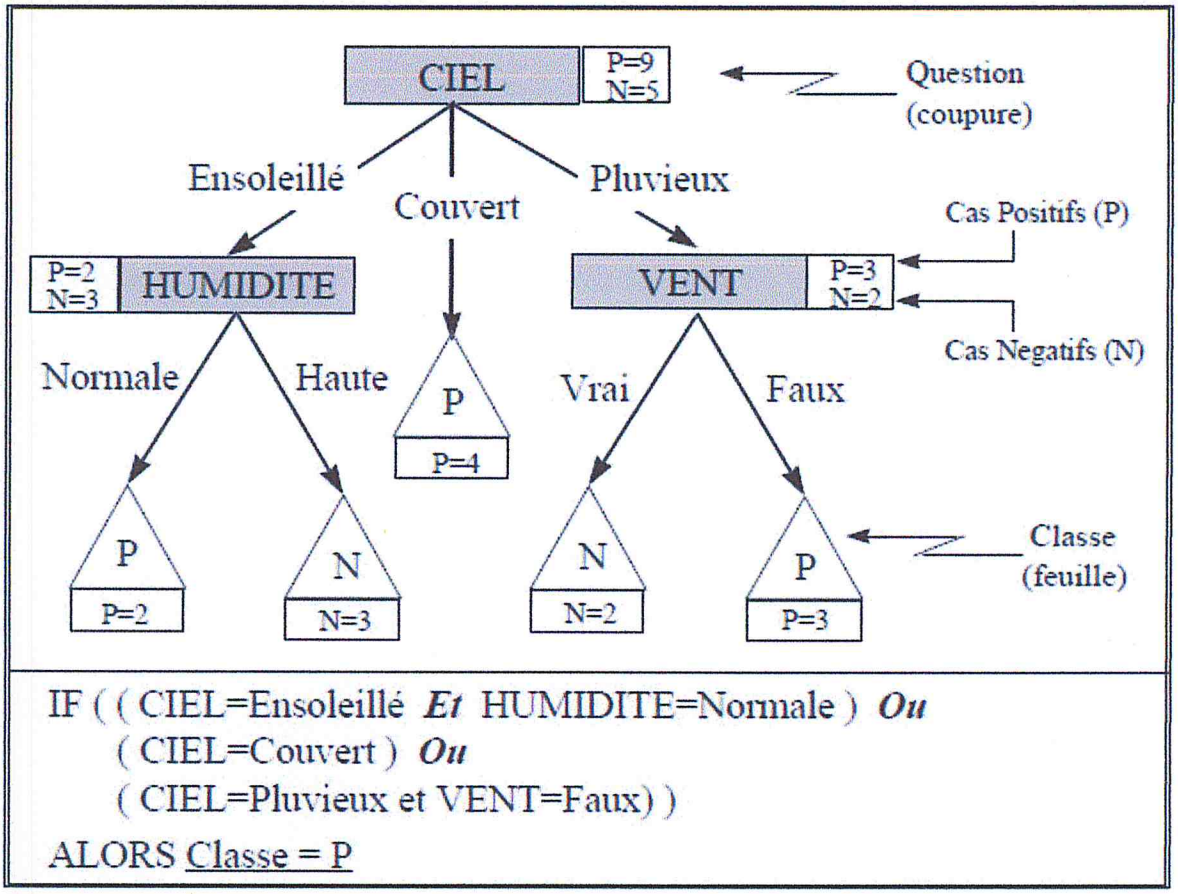


Fig 28 : Exemple arbre de décision simple [39]

L'apprentissage est fait à partir d'une base d'exemples qui possèdent un certain nombre d'attributs significatifs, e.g. la température, le vent, l'humidité, etc. Chaque exemple associe des valeurs particulières à chaque attribut, et comme cette méthode est une *méthode d'apprentissage supervisé*, chaque exemple est associé à une classe particulière.

2-2 Machine à vecteurs supports :

2-2-1 Définition :

Les machines à vecteurs de support, ou SVM (Support Vector Machines), sont une technique relativement récente (elles ont été introduites en 1992 par Vladimir Vapnik, Bernhard Boser et Isabelle Guyon) de classification supervisée qui suscite beaucoup d'intérêt pour ses bonnes performances dans un large éventail d'applications pratiques [48].

Les machines à vecteurs de support (SVM) sont un algorithme dont le but est de résoudre les problèmes de discrimination à deux classes. On appelle problème de

discrimination à deux classes un problème dans lequel on tente de déterminer la classe à laquelle appartient un individu (individu est ici employé au sens de constituant d'un ensemble) parmi deux choix possibles.

Pour ce faire, on utilise les caractéristiques connues de cet individu. Ces n caractéristiques sont représentés par un vecteur $x \in R^n$. La classe à laquelle appartient l'individu est représentée par $y \in \{-1, 1\}$, où une des classes possible est représentée par -1 et l'autre par 1. Par conséquent, avec cette notation, le problème est de déterminer la valeur de « y » en se servant de « x ». Pour y parvenir, les machines à vecteurs de support utilisent un ensemble de données pour lesquelles le classement est déjà connu et s'en servent pour construire une règle qui permet d'effectuer une bonne classification. Cet ensemble de données est appelé l'ensemble d'apprentissage. La règle trouvée avec l'ensemble d'apprentissage doit être la plus générale possible, puisqu'il faut aussi qu'elle soit bonne pour de nouvelles données qui n'étaient pas dans l'ensemble d'apprentissage.

2-2-2 Hyperplan séparateur :

Supposons que nous disposons d'un ensemble d'apprentissage de l données de la forme $(x_i, y_i) \in R^n \times \{-1,1\}$ (i = 1 ,.., l), dont nous voulons nous servir pour déterminer une règle permettant de classer les données. Supposons aussi que ces données sont linéairement séparables, c'est-à-dire qu'il existe un hyperplan dans R^n tel que toutes les données appartenant à la classe 1 se retrouvent d'un côté de l'hyperplan alors que celles de la classe -1 se situent de l'autre côté [48].

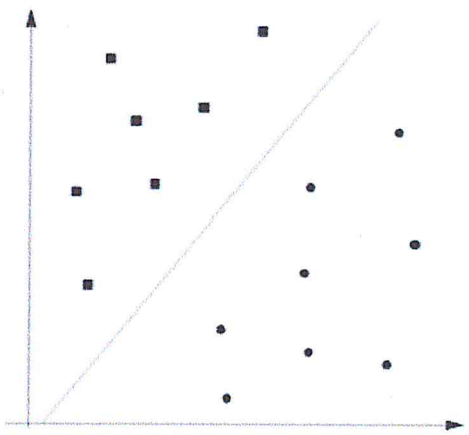


Fig 29 : Des données linéairement séparables [48]

Plus formellement, les données sont dites linéairement séparables s'il existe un hyperplan $w.x + b = 0$ tel que $w.x + b > 0$ pour tout x appartenant à la classe 1, et $w.x + b < 0$ pour tout x appartenant à la classe -1, avec $w = (w_1, \ldots, w_n) \in R^n$ le vecteur des coefficients de l'hyperplan et $b \in R$ un scalaire appelé le biais [48]. Sous l'hypothèse que les données sont linéairement séparables, trouver une règle pour les classer est très simple. En effet, il suffit de prendre un hyperplan qui sépare les classes, puis de classer les données selon le côté de l'hyperplan où elles se trouvent. Plus formellement, soit $w.x + b = 0$ un hyperplan qui sépare les données. Alors, il suffit d'utiliser la fonction suivante (parfois appelée la fonction indicatrice) pour effectuer la classification : Classe (x) = signe $(w.x + b)$

Où

$$\text{signe}(\mathrm{w} \cdot \mathrm{x} + b) = \begin{cases} -1 & \text{si } \mathrm{w} \cdot \mathrm{x} + b < 0 \\ 0 & \text{si } \mathrm{w} \cdot \mathrm{x} + b = 0 \\ 1 & \text{si } \mathrm{w} \cdot \mathrm{x} + b > 0 \end{cases}$$

Cette fonction classe les données par rapport au côté de l'hyperplan où elles se trouvent. On remarque que si un ensemble de données est séparé par un hyperplan, il sera parfaitement classé par cette fonction. Notons que si une donnée est directement sur l'hyperplan (ce qui peut arriver en considérant des données qui ne sont pas dans l'ensemble d'apprentissage), elle sera assignée à la classe 0, ce qui signifie qu'elle ne peut être classée par le modèle actuel. Dans ce cas, il est possible de la laisser inclassée, d'utiliser une autre règle ou de l'assigner aléatoirement à l'une des deux classes [48].

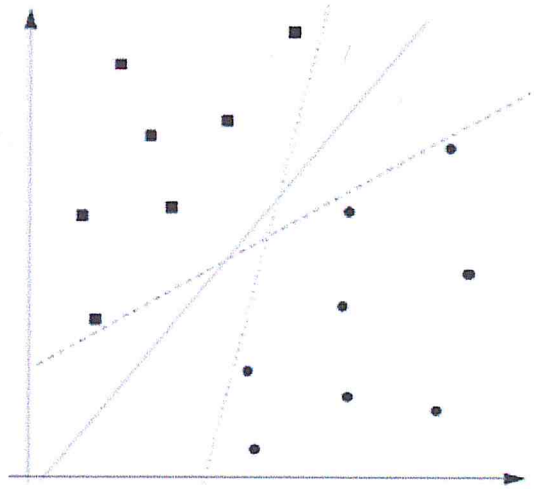


Fig 30 : Il existe une infinité d'hyperplans pouvant séparer les données [48]

Grâce à la fonction indicatrice, on constate qu'il suffit de trouver un hyperplan qui sépare les données pour déterminer une règle permettant de les classer. Cependant, si les données sont linéairement séparables, 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.

2-3 Algorithme des k plus proches voisins :

L'algorithme des k-plus-proches-voisins est l'un des algorithmes les plus simples d'apprentissage automatique supervisé. En supposant qu'une base d'apprentissage correctement étiquetée soit à disposition, cette méthode permet d'obtenir de très bons résultats de classification.

La méthode des plus proches voisins (noté parfois k-PPV ou k-NN pour k-Neighrest-Neighbors en anglais) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, la plus classique est la distance euclidienne, et le nombre de voisins à prendre en compte [62].

Les étapes de l'algorithme sont comme suit :

1. initialisation, choix de :
    - Nombre de classes, Valeur de k, exemples initiaux (base d'apprentissage), mesure de similarité.
2. pour chaque vecteur d'objet à classer :
    - mesurer la distance du vecteur avec tous les autres déjà classés
    - déterminer la liste des k vecteurs les plus proches de lui (k-ppv)
    - déterminer la classe la plus représentée dans la liste des k-ppv et affecter notre vecteur à cette classe.

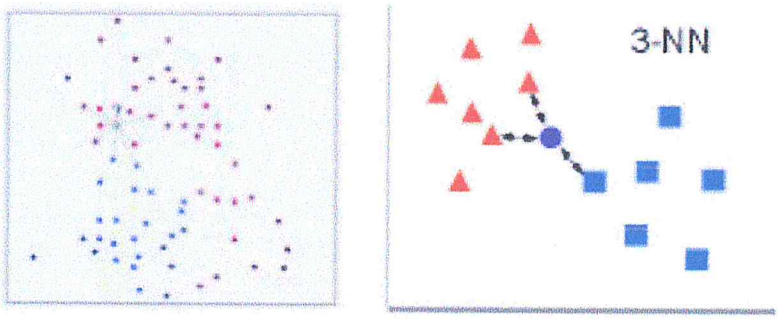


Fig 31 : Exemple illustratif [62]

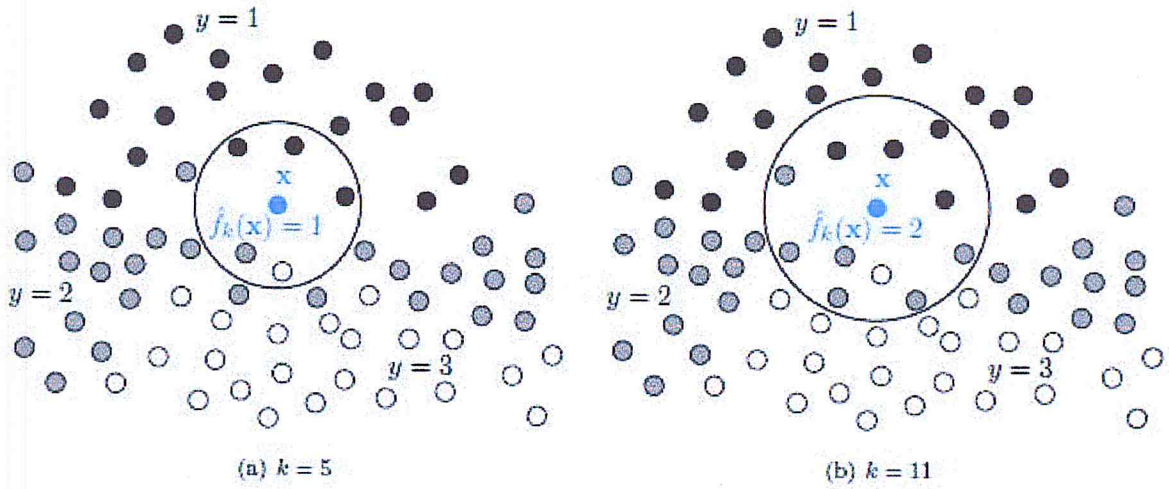


Fig 32 : Exemple de fonctionnement de la méthode des k-plus proches voisins pour des valeurs du paramètres k = 5 et k = 11. On considère trois classes, représentées respectivement en noir ($y = 1$), en gris ($y = 2$) et en blanc ($y = 3$). [62]

3- Réseaux de neurones :

3-1 Définition :

Les réseaux connexionnistes ou réseaux de neurones sont des assemblages fortement connectés d'unités de calcul. Ces derniers ont pour origine un modèle du *neurone biologique*, dont ils ne retiennent d'ailleurs qu'une vision fort simplifiée (voir Figure.33 et Figure.34). Le neurone, comme toute cellule, est composé d'un corps (ou *soma*), qui contient son noyau et où se déroulent les activités propres à sa vie cellulaire. Cependant, il est aussi doté d'un *axone* et de *dendrites*, structures spécialisées dans la communication avec les autres neurones. Cette communication entre cellules nerveuses s'effectue via des impulsions nerveuses. Les impulsions sont générées à l'extrémité

somatique de l'axone et vont vers les terminaisons axonales. Là, elles affecteront tous les neurones reliés au neurone générateur, par l'intermédiaire de jonctions entre les terminaisons axonales et les autres cellules. Cette jonction est appelée *synapse*. [39]

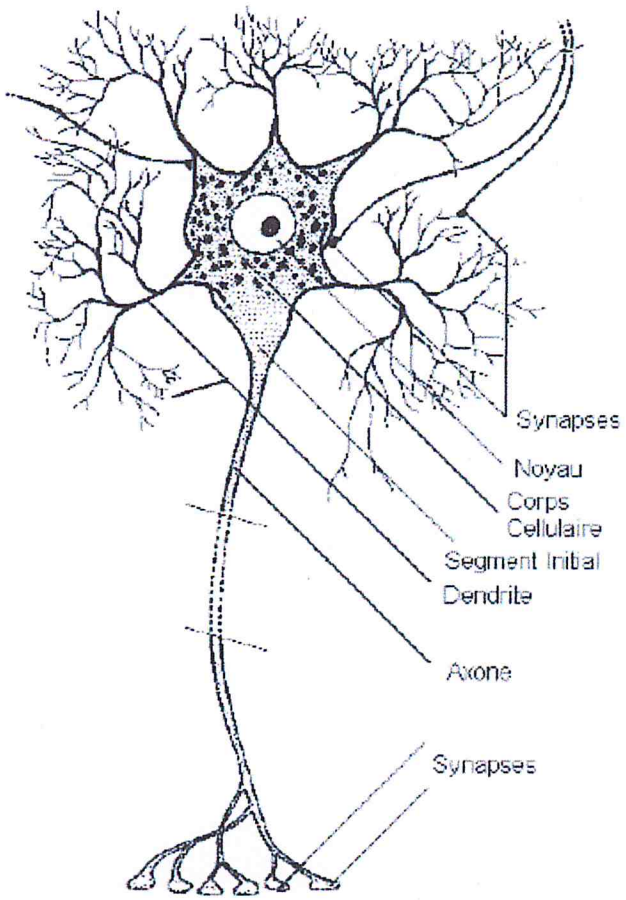


Fig 33 : Exemple de Neurone Biologique [39]

Cet héritage de la neurobiologie forme une composante importante de l'étude des réseaux connexionnistes, et le souci de maintenir une certaine correspondance avec le système nerveux humain a animé une part importante des recherches dans ce domaine. Malgré cet héritage, l'essentiel des travaux d'aujourd'hui ont pour objet les réseaux de neurones formels qui possèdent plusieurs propriétés, et qui les rendent intéressants d'un point de vue théorique, et fort utiles en pratique.

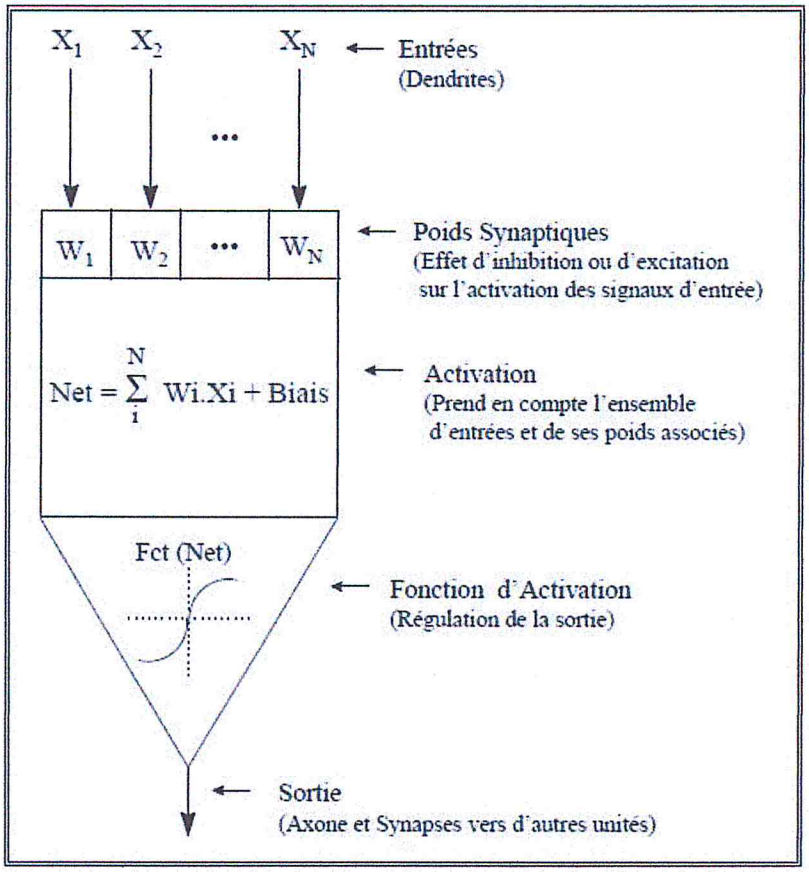


Fig 34 : Exemple de Neurone formel [54]

Un réseau neuronal est l'association, en un graphe plus ou moins complexe, d'objets élémentaires « *les neurones formels* ». Les principaux réseaux se distinguent par l'organisation du graphe, c'est-à-dire leur architecture, son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), par le type des neurones (leurs fonctions de transition ou d'activation) et enfin par l'objectif visé : apprentissage supervisé ou non, optimisation, systèmes dynamiques [54].

3-2 Neurone formel :

Par analogie au neurone biologique, le neurone formel est un modèle qui se caractérise par un état interne s∈S, des signaux d'entrée $x_1,x_2,\ldots,x_p$ et une fonction d'activation.

$$s = h(x_1, \ldots, x_p) = f(\alpha_0 + \sum_{j=1}^{p} \alpha_j x_j)$$

La fonction d'activation opère une transformation d'une combinaison affine des signaux d'entrée, α_0 étant appelé le biais du neurone. Cette combinaison affine est déterminée

par un vecteur de poids $[\alpha_0, \ldots., \alpha_p]$ associé à chaque neurone et dont les valeurs sont estimées dans la phase d'apprentissage.

Les différents types de neurones se distinguent par la nature f de leur fonction d'activation. Les principaux types sont :

- Linéaire : f est la fonction identité
- Sigmoïde : $f(x)=1 / (1 + e^x)$
- Seuil : $f(x) = 1_{[0,+\infty[}(x)$
- Radiale : $f(x) = \sqrt{1/2\pi \exp(-x^2/2}$

Les modèles linéaires et sigmoïdaux sont bien adaptés aux algorithmes d'apprentissage impliquant une rétro-propagation du gradient car leur fonction d'activation est différentiable ; ce sont les plus utilisés [54].

3-2.1 Perceptron multicouches :

Le perceptron multicouche (PMC ou MLP pour Multi-Layer-Perceptron) est un réseau composé de couches successives. Une couche est un ensemble de neurones n'ayant pas de connexion entre eux. Une couche d'entrée lit les signaux entrant, un neurone par entrée x_j, une couche en sortie fournit la réponse du système. [57]

Une ou plusieurs couches cachées participent au transfert. Un neurone d'une couche cachée est connecté en entrée à chacun des neurones de la couche précédente et en sortie à chaque neurone de la couche suivante.

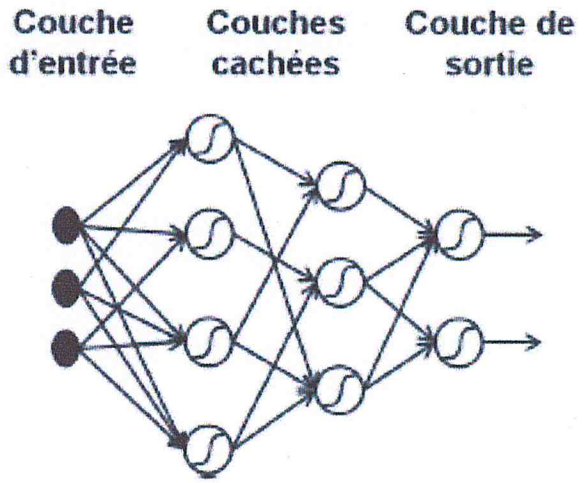


Fig 35 : Exemple de perceptron multicouche avec une couche d'entrée, deux couches cachée et une couche de sortie [57]

$\sigma_i(x)$: Désignera la fonction d'activation correspondant à un neurone *i*.

Si nous considérons un Perceptron multicouches avec N neurones d'entrée, activés par un vecteur d'entrée x (de taille N), et par $w_{ij}^{0,1}$ le poids correspondant à la connexion entre le neurone *i* de la couche 0 et le neurone *j* de la couche 1, la sortie a_j^1 de chacun des neurones de la première couche cachée sera exprimée par : [29]

$$a_j^1 = \sigma_j(b_j^1 + \sum_{i=1}^{N} w_{ij}^{0,1} x_i) \qquad \text{"Equation 1"}$$

Où σ_i est la fonction d'activation décrite précédemment, et b_j^1 est un paramètre supplémentaire appelé biais, qui peut être considéré comme le poids d'une entrée constante égale à 1, et dont le rôle est de rajouter un degré de liberté supplémentaire en agissant sur la position de la frontière de décision.

Ce même processus exprimé par l'équation 1 peut être répété pour les autres couches (cachées ou celle de sortie) : Chaque sortie d'une couche *l* joue le rôle d'entrée pour la couche suivante $l + 1$. Ainsi, nous pouvons généraliser l'équation 1 à toutes les couches suivantes (y compris la couche de sortie) comme suit : [57]

$$a_j^{l+1} = \sigma_j(b_j^{l+1} + \sum_{i=1}^{L} w_{ij}^{l,l+1}\ a_i^l) \qquad \text{"Equation 2"}$$

Où « L » est le nombre de neurones de la couche *l*.

Les Perceptrons multicouches sont généralement utilisés pour des problématiques de classification supervisée. Ceci implique l'existence d'un ensemble de paires d'entrées sorties (appelé base d'apprentissage) liés par une certaine relation, que le réseau va "apprendre" en ajustant ses paramètres.

3-2.2 Réseaux de neurones à convolution (CNN):

3-2.2.1 Définition :

Les réseaux neuronaux convolutionnels (ConvNets ou CNN) sont une catégorie de réseaux neuronaux qui se sont révélés très efficaces dans des domaines tels que la reconnaissance et la classification de l'image. ConvNets a réussi à identifier les visages, les objets et les panneaux de signalisation en dehors de l'alimentation de la vision dans les robots et les voitures auto-conductrices [69].

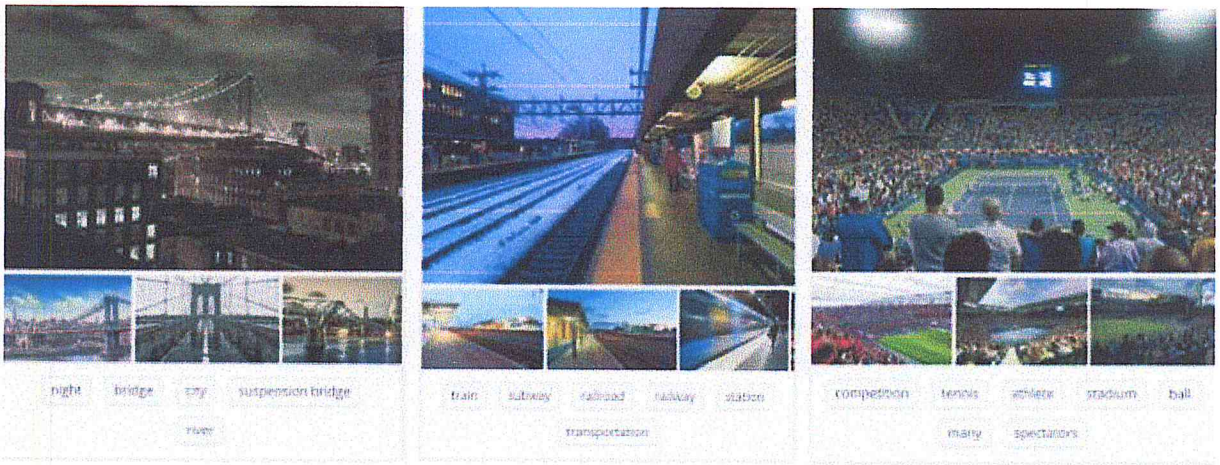


Fig 36: ConvNet pour la reconnaissance des scènes [69]

Fig 37: ConvNet pour la reconnaissance des objets [69]

Dans la figure36 ci-dessus, un ConvNet est capable de reconnaître des scènes, «pont», «chemin de fer» et «tennis» tandis que la figure37 montre un exemple de ConvNets utilisé pour reconnaître les objets de tous les jours, les humains et les animaux.

ConvNets est donc un outil important pour la plupart des praticiens de l'apprentissage en machine aujourd'hui.

3-2.2.2 L'architecture LeNet (1990) :

LeNet a été l'un des premiers réseaux neuronaux convolutifs qui ont aidé à propulser le domaine de l'Apprentissage Profond. Le travail de Yann LeCun a été nommé LeNet5 après de nombreuses précédentes itérations réussies depuis 1988. À cette époque,

l'architecture LeNet était principalement utilisée pour les tâches de reconnaissance de caractères telles que la lecture des codes postaux, des chiffres, etc.[69]

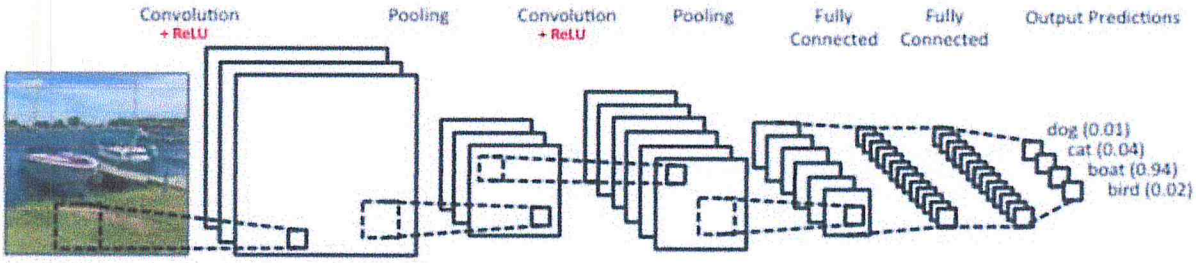


Fig 38 : un ConvNet simple [69]

Le réseau neuronal convolutif de la figure38 est similaire en architecture au LeNet d'origine et classe une image d'entrée en quatre catégories: chien, chat, bateau ou oiseau (le LeNet original a été utilisé principalement pour les tâches de reconnaissance de caractères). Comme la montre la figure ci-dessus, en recevant une image de bateau comme entrée, le réseau attribue correctement la plus grande probabilité de bateau (0,94) parmi les quatre catégories. La somme de toutes les probabilités dans la couche de sortie devrait être 1.

Il existe quatre opérations principales dans le ConvNet, illustrées à la figure38 ci-dessus:

1. Convolution
2. Non linéarité (ReLU)
3. Pooling ou Sub Sampling
4. Classification (couche entièrement connectée)

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

3-2.2.3 L'étape de la convolution :

Les ConvNets tirent leur nom de l'opérateur "convolution". Le but principal de Convolution dans le cas d'un ConvNet est d'extraire des fonctionnalités de l'image d'entrée. La convolution préserve la relation spatiale entre les pixels en apprenant des caractéristiques d'image à l'aide de petits carrés de données d'entrée. [69]

Chaque image peut être considérée comme une matrice de valeurs de pixels. Considérons une image 5 x 5 dont les valeurs de pixel sont seulement 0 et 1 (notez que pour une image en niveaux de gris, les valeurs de pixels vont de 0 à 255, la matrice verte ci-dessous est un cas particulier où les valeurs de pixel sont seulement 0 et 1):

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Aussi, considérez une autre matrice 3 x 3 comme indiqué ci-dessous:

1	0	1
0	1	0
1	0	1

Ensuite, la Convolution de l'image 5 x 5 et la matrice 3 x 3 peuvent être calculées comme indiqué dans la Figure 39 ci-dessous:

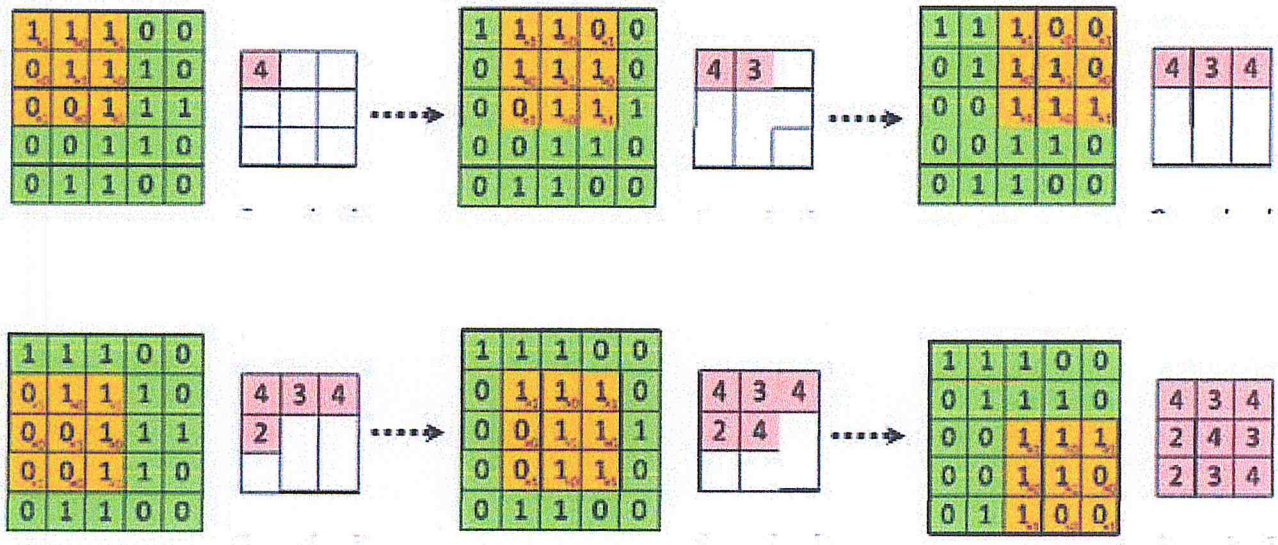


Fig 39: résultat de la convolution [69]

Le figure 39 présente l'opération de convolution. La matrice de sortie s'appelle Convolved Feature ou **Feature Map.**

Nous glissons la matrice orange sur l'image originale (vert) de 1 pixel (également appelé 'stride') et pour chaque position, nous calculons la multiplication des éléments (entre les deux matrices) et ajoutons les sorties de multiplication pour obtenir l'entier final qui se forme Un seul élément de la matrice de sortie (rose). Notez que la matrice 3 × 3 "ne voit" qu'une partie de l'image d'entrée à chaque étape.[69]

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.[69]

Considérez l'image d'entrée suivante:

Dans le tableau ci-dessous, nous pouvons voir les effets de la convolution de l'image ci-dessus avec différents filtres. Comme indiqué, nous pouvons effectuer des opérations telles que la détection de bordure, la netteté et le flou simplement en changeant les valeurs numériques de notre matrice de filtre avant l'opération de convolution - cela signifie que différents filtres peuvent détecter différentes caractéristiques d'une image, par exemple des bords, Des courbes, etc.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Tableau 1: les effets de la convolution de l'image ci-dessus avec différents filtres [69]

En pratique, une CNN apprend les valeurs de ces filtres tout seul pendant le processus de formation (bien qu'il soit nécessaire de spécifier des paramètres tels que le nombre de filtres, la taille du filtre, l'architecture du réseau, etc. avant le processus de formation). Plus il y a de filtres, plus d'images sont extraites

La taille de Feature Map (Convolved Feature) est contrôlée par trois paramètres:

- **Profondeur (Depth):** la profondeur correspond au nombre de filtres que nous utilisons pour l'opération de convolution. Dans le réseau représenté à la figure 40, une

convolution de l'image originale du bateau en utilisant trois filtres distincts, produisant ainsi trois cartes de caractéristiques différentes, comme illustré.

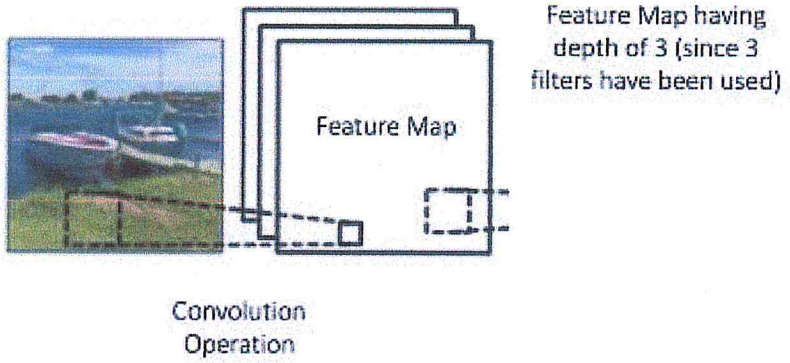


Fig 40 : Features Map obtenus avec 3 filtres [69]

- **Stride:** Stride est le nombre de pixels par lequel on glisse notre matrice de filtre sur la matrice d'entrée. Lorsque la foulée est 1, on déplace les filtres un pixel à la fois. Lorsque la foulée est 2, les filtres sautent 2 pixels à la fois alors que nous les glissons. Une plus grande foulée produira des cartes de fonctionnalités plus petites.[69]
- **Zéro-rembourrage** (Zero-padding)**:** Une caractéristique intéressante de la remise à zéro est qu'elle nous permet de contrôler la taille des cartes des fonctionnalités. L'ajout de zéro-rembourrage est également appelé une convolution large, et l'utilisation de zéro-rembourrage serait une convolution étroite. [69]

3-2.2.4 Présentation de la non linéarité (ReLU) :

Une opération supplémentaire appelée ReLU a été utilisée après chaque opération de convolution à la figure 38. ReLU signifie Unité linéaire rectifiée et est une opération non linéaire. Sa sortie est donnée par:

Output = Max(zero, Input)

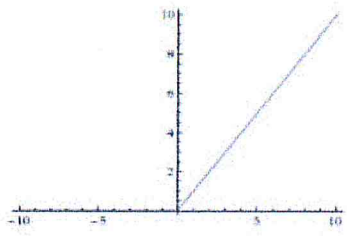


Fig 41 : opération de ReLU [69]

ReLU est une opération qui remplace toutes les valeurs de pixel négatives dans la carte des caractéristiques par zéro.

D'autres fonctions non linéaires telles que tanh ou sigmoid peuvent également être utilisées à la place de ReLU.

3-2.2.5 The Pooling Step :

Le Pooling spatiale réduit la dimensionnalité de chaque carte de caractéristiques, mais conserve les informations les plus importantes. Il peut être de différents types: Max, Moyenne, Somme, etc.

Dans le cas de Max Pooling, on prend le plus grand élément à partir de la carte des caractéristiques. Au lieu de prendre le plus grand élément, nous pouvons également prendre la moyenne (Pooling moyen) ou la somme de tous les éléments.

La figure 42 montre un exemple de l'opération Max Pooling sur une carte de fonctionnalités rectifiées (obtenue après convolution + opération ReLU) en utilisant une fenêtre 2 × 2. [69]

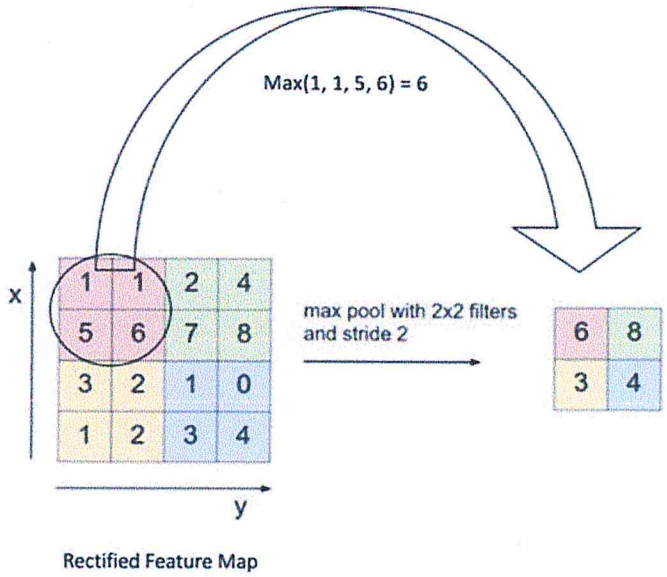


Fig 42: Max Pooling [69]

On glisse une fenêtre 2 x 2 par 2 cellules (également appelées 'stride') et on prend la valeur maximale dans chaque région. Comme le montre la figure 42, cela réduit la dimensionnalité de notre carte de caractéristiques.

Dans le réseau représenté à la Figure 43, l'opération de Pooling est appliquée séparément à chaque carte de caractéristiques (notez qu'en raison de cela, nous obtenons trois cartes de sortie à partir de trois cartes d'entrée).

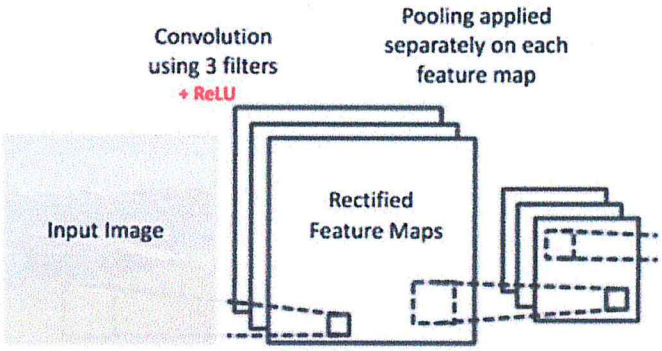


Fig 43: Max Pooling appliqué au Feature Maps [69]

La figure 44 montre l'effet de Pooling sur la carte de caractéristiques rectifiées après l'opération ReLU .

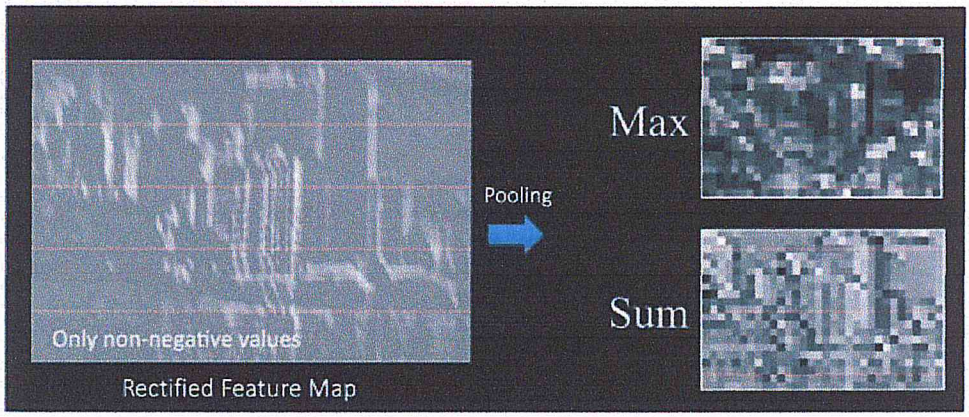


Fig 44: Max Pooling et Sum Pooling [69]

La fonction de Pooling consiste à réduire progressivement la taille spatiale de la représentation d'entrée [69]. En particulier, pooling :

- Rend les représentations d'entrée (dimension de la caractéristique) plus petites et plus faciles à gérer
- Réduit le nombre de paramètres et de calculs dans le réseau.
- Rend le réseau invariant à de petites transformations.
- Nous aide à arriver à une représentation invariable de l'image. Ceci est très puissant car nous pouvons détecter des objets dans une image.

3-2.3 Réseaux de neurones récurrents (RNN):

Les Perceptrons multi-couches, 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).[57]

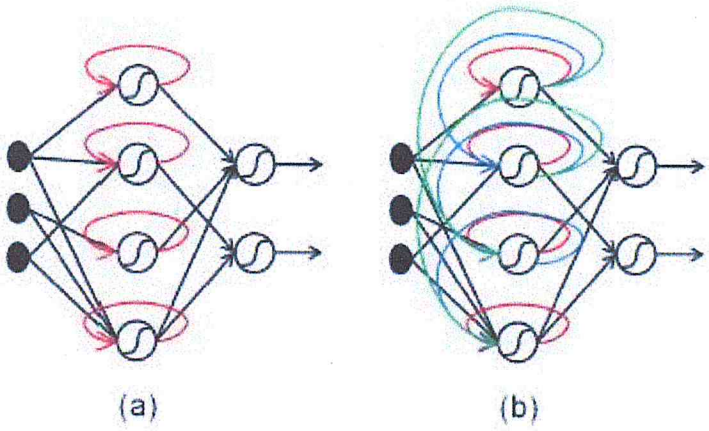


Fig 45 : Réseau de neurones récurrent avec une couche cachée : (a)- Réseau auto-récurrent basique (b)- Réseau récurrent totalement connecté. [57]

La Figure 45-(a) illustre le réseau de neurones récurrent 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). La Figure 45-(b) présente quant à elle un exemple de réseau de neurones récurrent plus complexe (dit totalement connecté), où tous les neurones de la couche cachée sont connectés entre eux. [57]

Plusieurs architectures récurrentes ont été définies dont le principe commun est d'apprendre une correspondance entre des séquences de vecteurs d'entrée, et des séquences de vecteurs désirés, en utilisant les connections récurrentes qui permettent de se "rappeler" d'un certain nombre d'états passés. Ainsi, à un instant t pour une séquence donnée, les RNNs font intervenir les instants passés lors du calcul de l'état présent.

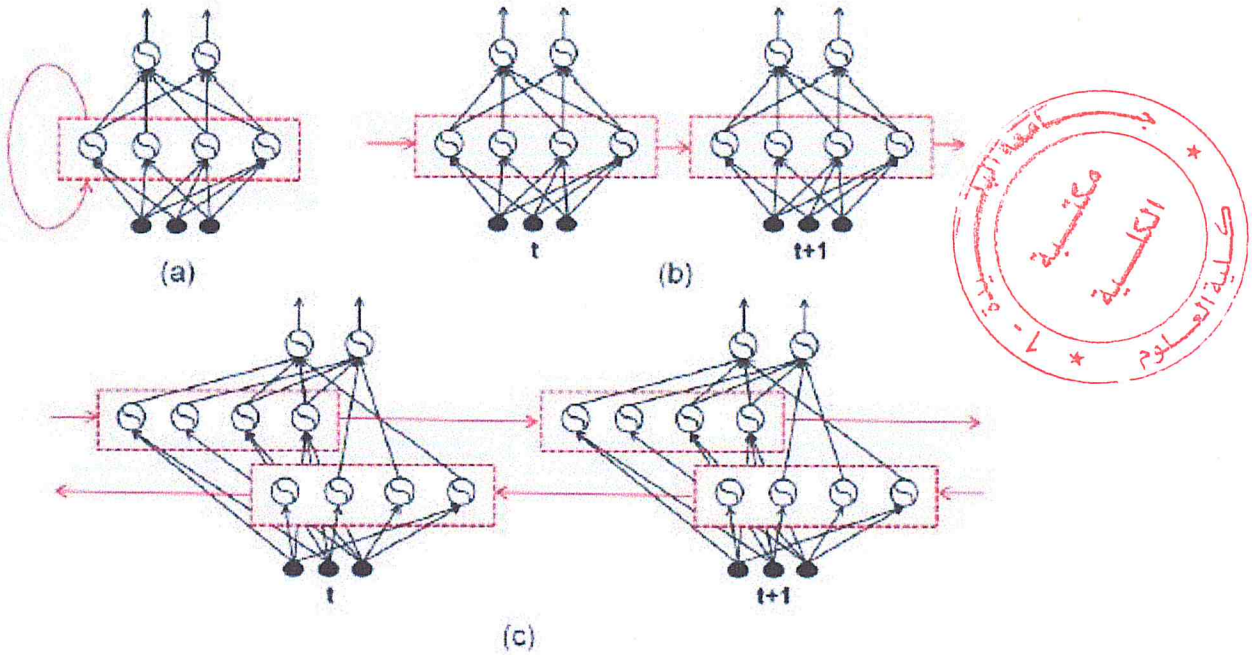


Fig 46 : (a)- Réseau récurrent unidirectionnel (b)- Réseau récurrent unidirectionnel en vue éclatée (c) - Réseau récurrent bidirectionnel en vue éclatée. [57]

La Figure 46-(a) présente un réseau récurrent unidirectionnel classique comme ceux décrits précédemment. Ce réseau peut être vu comme une succession de MLPs (un réseau pour chaque instant), avec des entrées classiques, mais aussi les sorties de la couche cachée du MLP correspondant à l'instant précédent. Ce principe est illustré sur la Figure 46-(b) et est appelé vue éclatée. [57]

L'un des moyens proposés dans la littérature pour augmenter la quantité d'information de contexte est de permettre à un instant t l'accès aussi bien au futur qu'au passé.

En pratique, ceci est fait en utilisant deux couches cachées : Une pour chaque direction. On parle alors de réseau récurrent bidirectionnel [15] (par opposition au réseau unidirectionnel présenté précédemment). La Figure 46-(c) représente la vue éclatée d'un réseau récurrent bidirectionnel : Les réseaux MLPs successifs comptent deux couches cachées (une pour chaque direction) qui sont connectées aux mêmes couches d'entrée et de sortie. Ces deux couches permettent, en théorie, au réseau à chaque instant d'avoir accès au contexte passé et futur d'une séquence donnée (tout se passe en fait comme si la séquence était présentée au réseau dans deux directions opposées).

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.

Chapitre IV

Approche proposé

Introduction :

La vidéosurveillance est un système de surveillance par des caméras qui peuvent être installées dans les espaces publics afin de gérer les risques en cas d'embouteillages, d'incendie, d'accident, d'actes criminels et comme résultat il va y'avoir un énorme flux de vidéos.

Ces énormes quantités de contenu vidéo ont largement dépassé la capacité que possède l'être humain de les visualiser et ont rendu difficile la recherche de contenus intéressants. La figure 50 représente un exemple illustratif dans lequel un réseau de caméras, capture des vidéos d'une région.

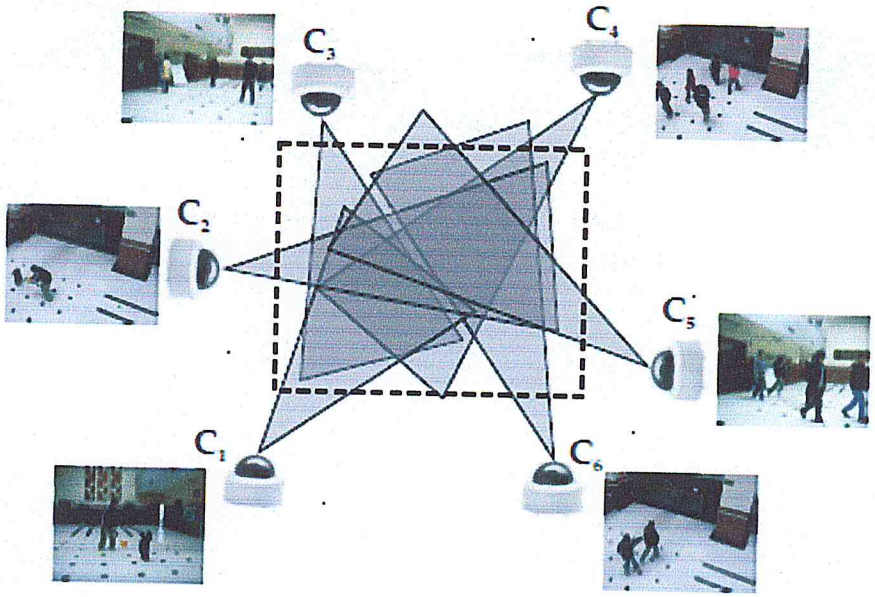


Fig 50 : Une illustration d'un réseau de caméras multi-sources où six caméras C1, C2, C3, C4, C5, C6 observe une zone (rectangle noir) à partir de différents angles. [67]

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.

Dans ce chapitre on va décrire notre approche qui est basé sur l'apprentissage profond afin de déterminer notre résumé vidéo tiré de plusieurs vues d'une même scène.

1. Détails de l'approche :

Le schéma suivant donne un aperçu général des différentes étapes pour la génération de notre résumé vidéo.

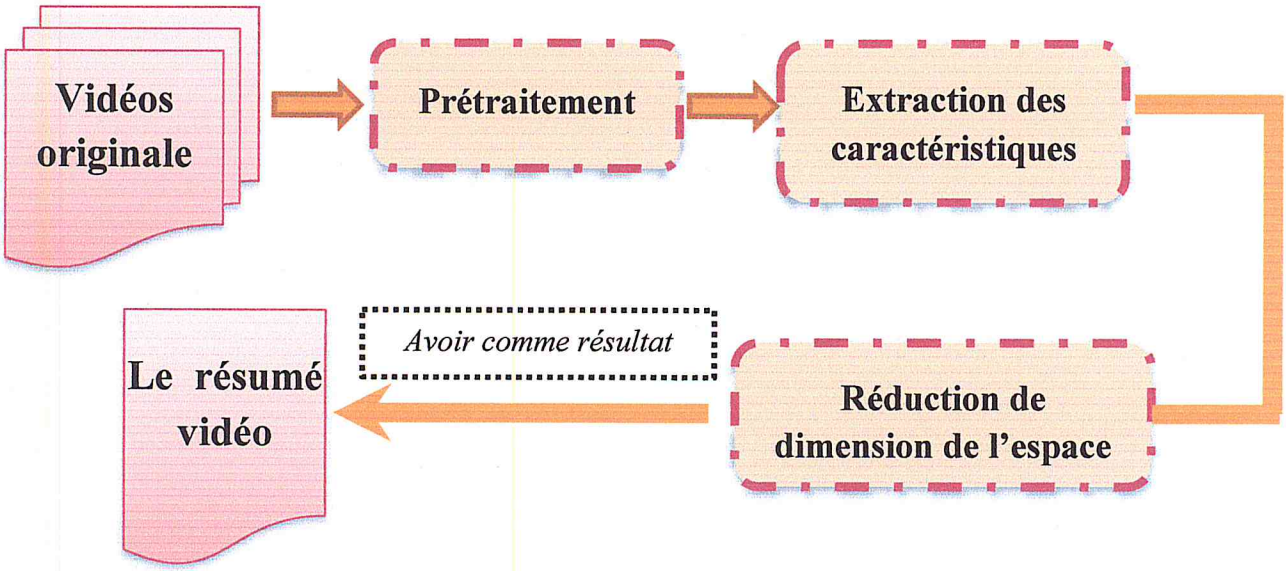


Figure 51 : schéma globale de notre approche

Considérons un ensemble de K différentes vidéos capturées à partir de différentes caméras, Set = $\{V_1, V_2, V_3, \ldots, V_K\}$.

Pour déterminer le « résumé vidéo » on doit passer par les différentes phases suivantes :

a) Phase de prétraitement
b) Phase d'extraction des caractéristiques
c) Phase de réduction de dimension de l'espace.

2. Phase de prétraitement :

Dans cette phase le travail qu'on doit effectuer consiste à extraire tous les trames pour chaque vidéo et la figure 52 illustre bien ces différentes étapes.

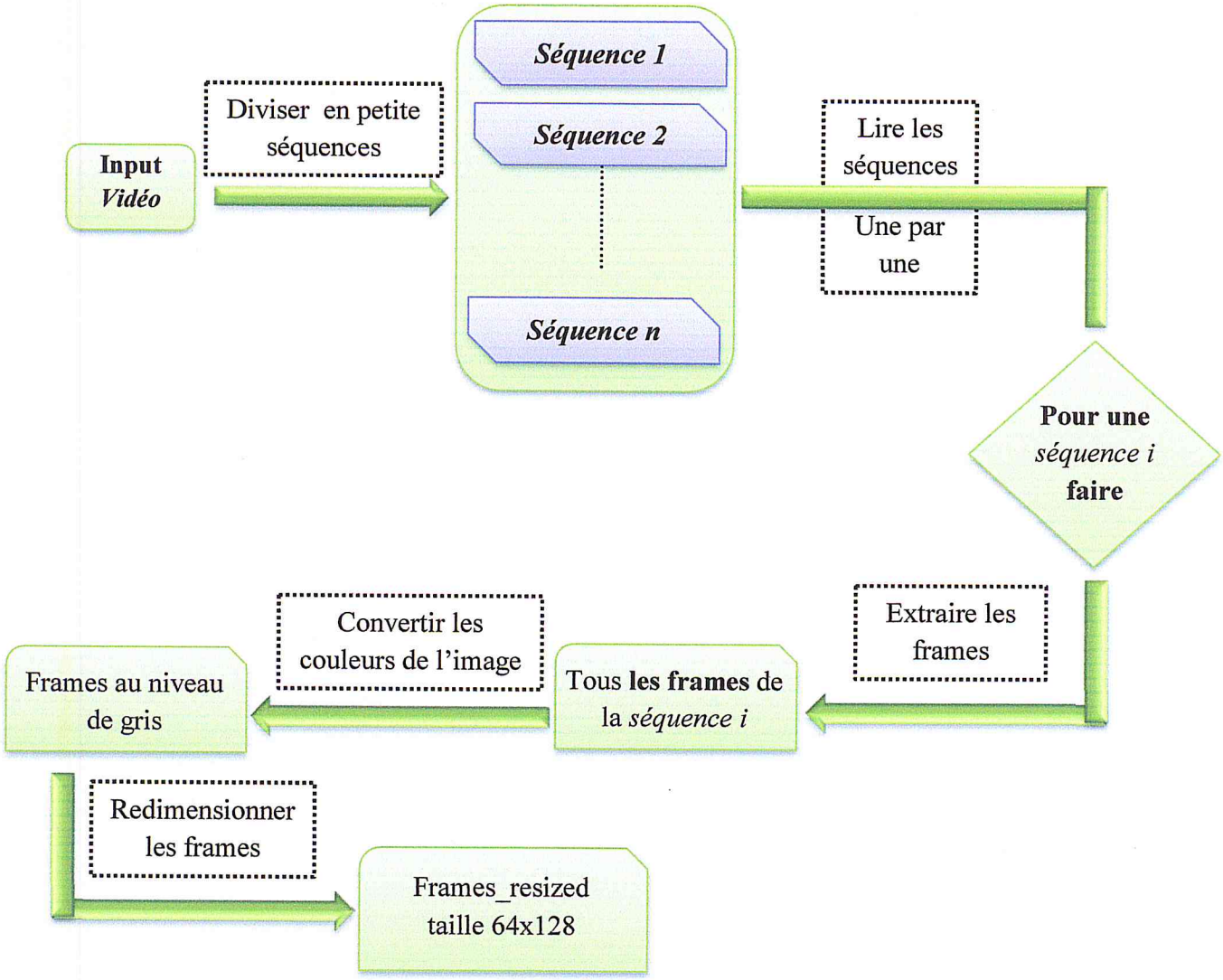


Fig 52 : schéma globale de la phase prétraitement

Les différentes étapes de la phase prétraitement :

1- Prendre une vidéo $Vidéo_i$ en entrée ;

2- Diviser cette *Vidéo* en petite séquences de quelques secondes (dans notre cas nous avons choisis une durée t *=20 secondes*) ;

3- Extraire toutes les frames de chaque séquence ;

4- Ensuite convertir les couleurs en niveau de gris ;

5- Redimensionner toutes les frames, dans notre étude nous avons choisi la taille de 64x128 (*frame_resized*$_i$, $i= 0..n$ où n est le nombre des trames de la séquence vidéo) ;

Les trames résultantes vont être introduite comme des « *inputs* » dans un module d'extraction de caractéristiques que nous allons le détaillé par la suite.
Le découpage de la vidéo en entrée en petite séquences garantie la corrélation intra-vidéo pour le calcul de la similarité entre les trames.

3. Phase d'extraction des caractéristiques :

Cette phase consiste à l'analyse du contenu des trames des séquences vidéo afin de représenter les données sous formes des vecteurs caractéristiques. La figure 53 montre un schéma global de cette phase.

La détermination de tous les vecteurs caractéristiques de la vidéo est réalisée en utilisant l'architecture CNN_BVLC (section 4) avec son modèle d'apprentissage profond pré-entrainé tel que :

$$X^{(k)} = \{x_i^{(k)} ; i=1,...,N_k\}, \text{k} = 1,...,\text{K}.$$

Chaque x_i représente le vecteur caractéristique d'une trame. Nous utilisons N_k pour désigner le nombre de trames dans la k-ième séquence vidéo et N pour désigner le nombre total d'images dans toutes les vidéos.

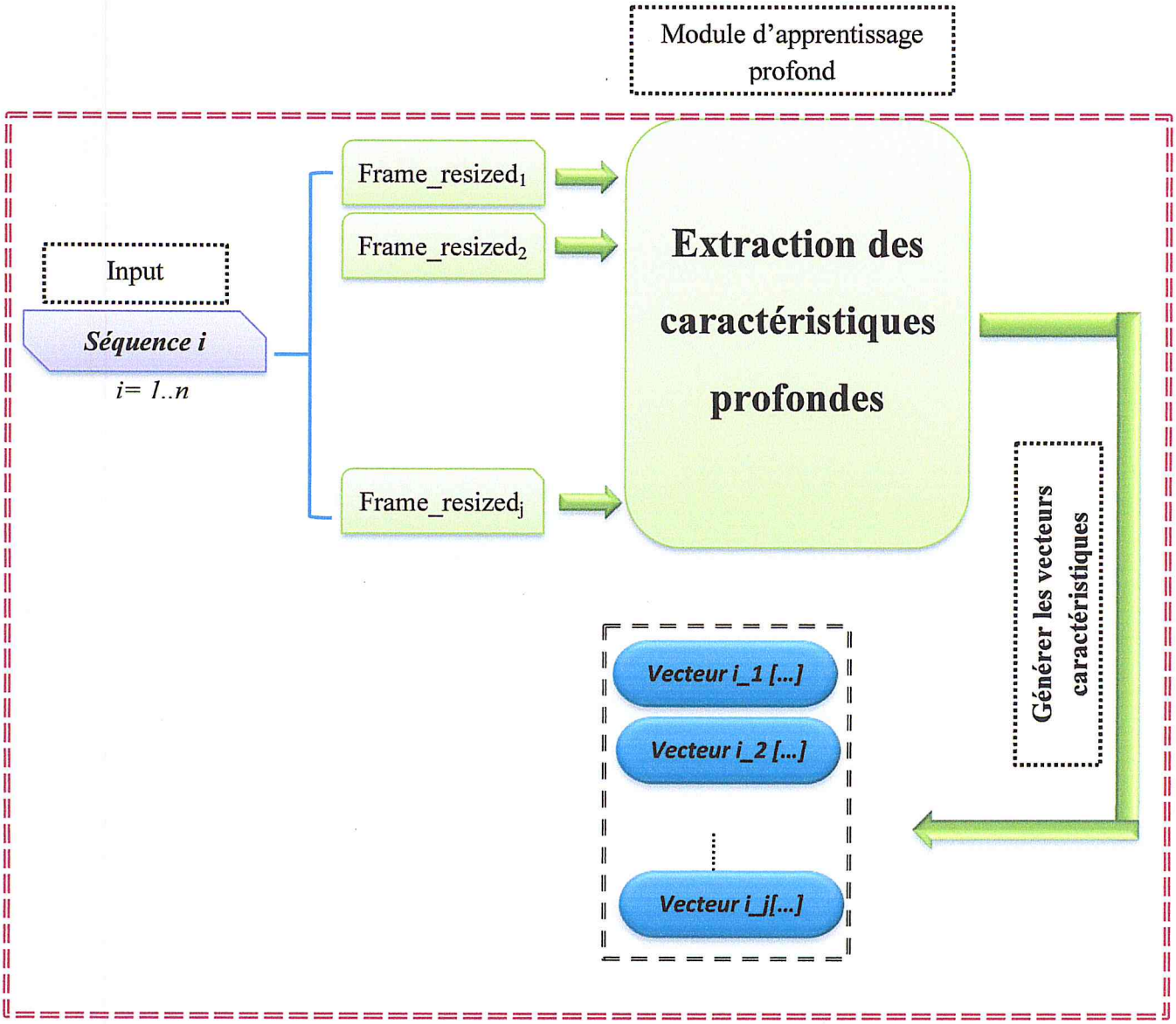


Fig 53 : schéma globale de la phase d'extraction des caractéristiques profondes

Après un certain nombre d'itérations on obtient tous les vecteurs caractéristiques des trames vidéo, la figure 54 montre la représentation d'une vidéo sous forme d'un ensemble de vecteurs caractéristiques.

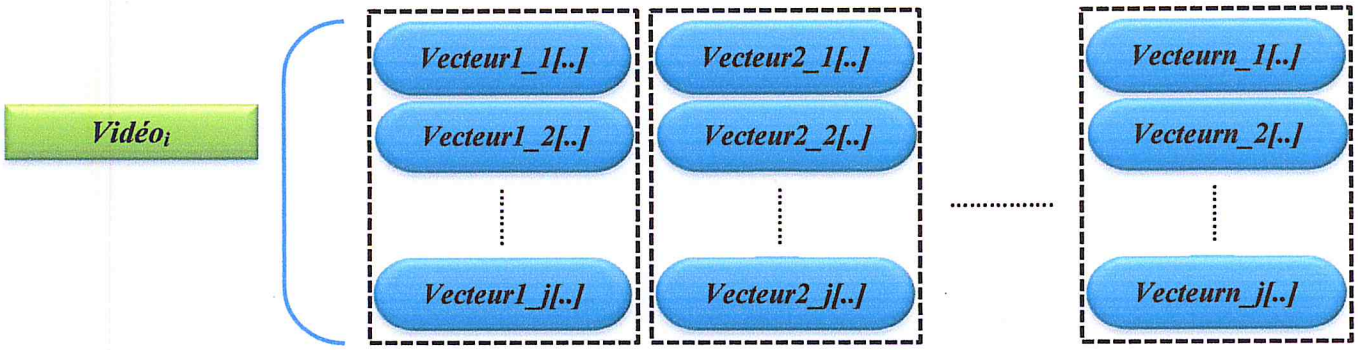


Fig 54 : Représentation de la vidéo sous formes d'un ensemble de vecteurs caractéristiques extrait dans la phase d'extraction de caractéristiques

4. Le modèle BVLC :

4-1 Définition :

L'architecture CNN_BVLC est un modèle de la vision par ordinateur conçu pour des tâches telles que la classification, la localisation et la détection d'objet.
L'architecture du réseau est composée de 5 couches de convolution, et trois couches de max-pooling. Ce réseau a été utilisé pour le classement avec 1000 catégories possibles.[71]

4-2 quelques caractéristique de BVLC:[71]

- Le réseau a été entrainé sur la base de données ImageNet, qui contient plus de 15 millions d'images annotées sur un total de plus de 22 000 catégories.
- ReLU est utilisé pour les fonctions de non-linéarité (pour diminuer le temps de formation car les ReLUs sont plusieurs fois plus rapides que la fonction Tanh conventionnelle).
- entrainé sur deux GPU GTX 580 pendant cinq à six jours

4-3 Architecture de BVLC:

Ce modèle mappe une image x_i d'entrée en couleur 2D, via une série de couches. Chaque couche consiste à [59] :

- Appliquer la convolution de la sortie de la couche précédente (ou, dans le cas de la 1ère couche, l'image d'entrée) avec un ensemble de filtres acquis;
- Passer les réponses par une fonction linéaire corrigée (relu (x) = max (x, 0));

- Les premières couches du réseau sont des réseaux conventionnels entièrement connectés et la couche finale est un classificateur de softmax. La figue 55 montre le modèle utilisé pour notre résumé.

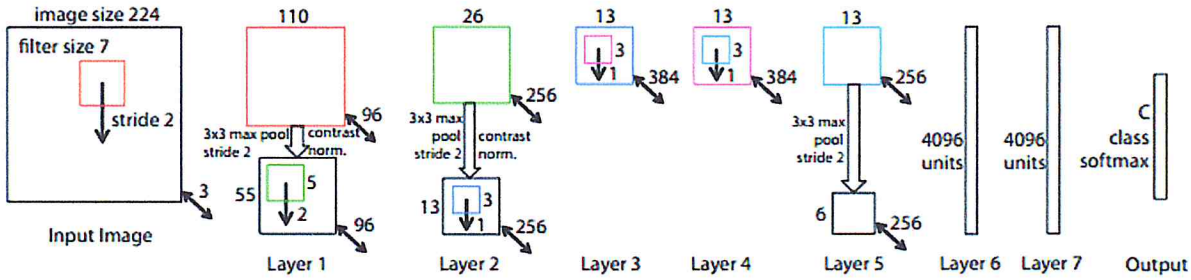


Fig 55 : Architecture de 8 couches du model BVLC [59]

Une image de 224 par 224 (avec 3 canaux de couleur) est présentée comme entrée. Celle ci est converti avec 96 différents filtres dans la 1ère couche (rouge), chacun de la taille 7 par 7, en utilisant un stride de 2 x 2. Les cartes caractéristiques qui en résultent sont alors: (i) passées à travers une fonction linéaire corrigée (non représentée), (ii) regroupées (pooled)(maximum dans les régions 3x3, à l'aide d'un stride 2) et dont le contraste est normalisé sur les cartes de caractéristiques pour donner 96 différents 55x55 cartes de caractéristiques d'élément. Des opérations similaires sont répétées dans les couches 2, 3, 4, 5. Les deux dernières couches sont entièrement connectées, en prenant des caractéristiques de la couche convolutionnelle supérieure comme entrée sous forme vectorielle ($6 \cdot 6 \cdot 256 = 9216$ dimensions). La couche finale est une fonction C-way softmax, C étant le nombre de classes. Tous les filtres et les cartes de caractéristiques sont de forme carrée.

La première couche du ConvNet est toujours un détecteur de caractéristiques de faible niveau qui détectera des bords ou des couleurs simples. Avec la deuxième couche, nous avons plus de fonctionnalités circulaires qui sont détectées. Comme le montre la Figure 56 [71].

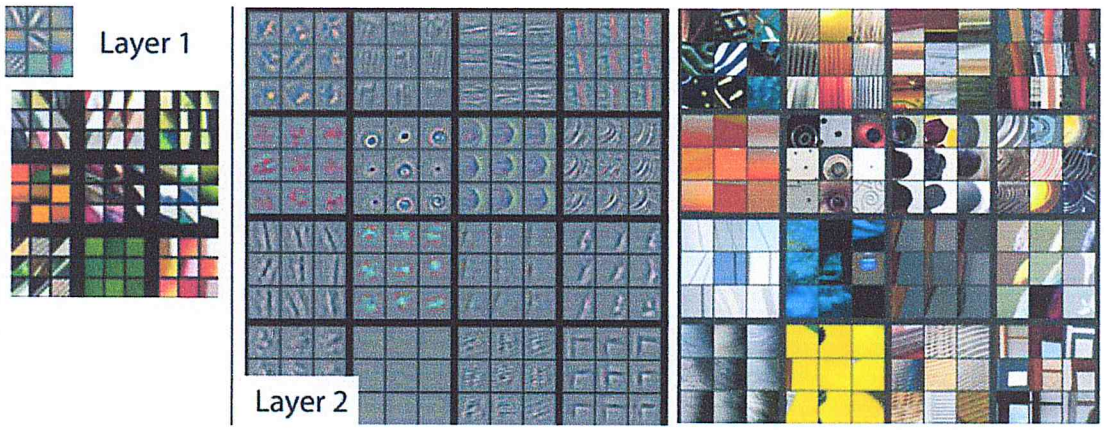


Fig 56 : visualization des couches 1 et 2 [59]

La figure 56 visualise les couches 1 et 2. Chaque couche est illustrée par deux images, une montre les filtres et l'autre montre quelle partie de l'image est fortement activée par le filtre donné.

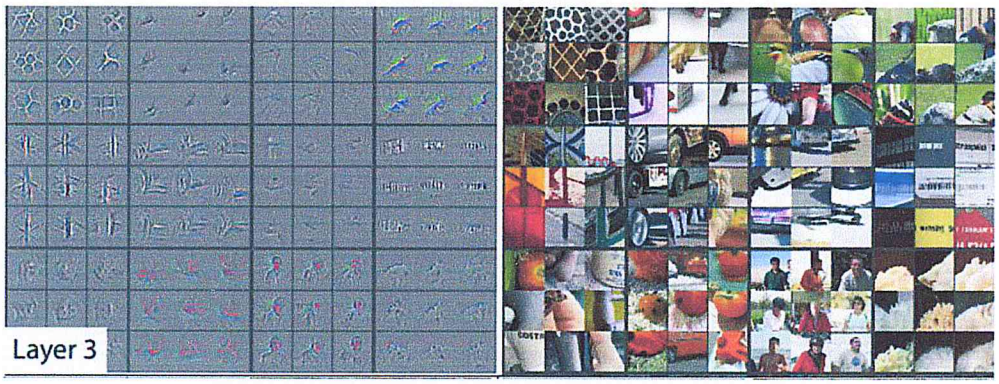


Fig 57 : visualization de la couche 3 [59]

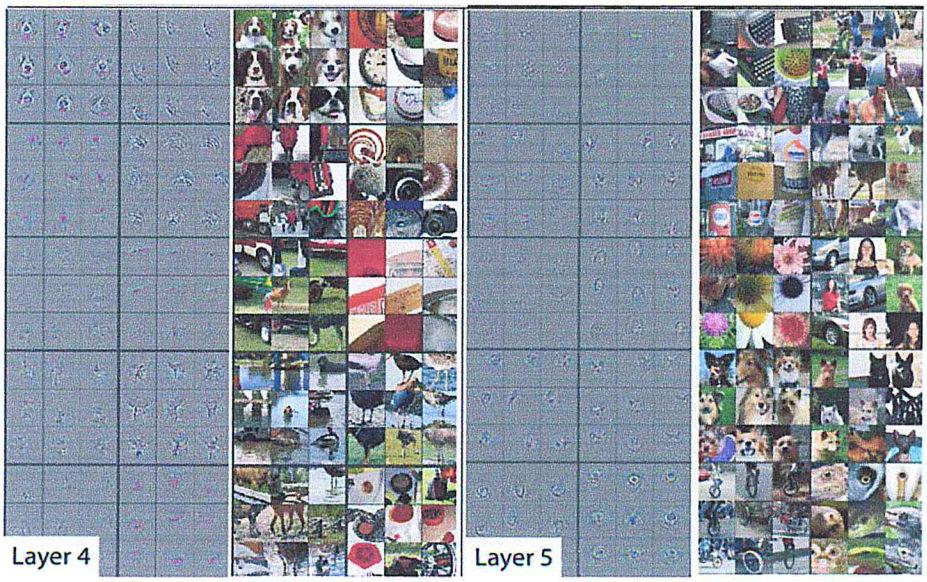


Fig 58 : visualization des couches 4 et 5 [59]

Les couches 3, 4, et 5 montrent beaucoup plus de caractéristiques de haut niveau telles que les visages, les fleurs ou des chiens.

On extraira les valeurs caractéristiques trouvées à la sixième couche.

5. **Phase de réduction de dimension de l'espace :**

Une fois que toutes les caractéristiques profondes sont extraites, 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 possible, en utilisant l'algorithme SNE pour Stochastic Neighbor Embedding [74] qui tente de représenter une donnée dans un espace a à haute dimension vers un espace à faible dimension.

Soit 'X' l'ensemble des vecteurs caractéristiques de dimension 4096, on veut représenter ces valeurs sur un espace plus réduit 'Y' de dimension d<<4096.

On cherche à trouver ce 'd' de telle sorte à ce que les similarités de 'Y' (matice Q) ont la même distribution que les similarités de 'X' (matrice P).

Où la matrice de distribution des similarités de X est calculée comme suit :

$$p_{ij} = \frac{e(-d_{ij}^2)}{\sum_{k \neq i} e(-d_{ik}^2)} \qquad (1)$$

d_{ij}^2 peut être calculées à l'aide de la distance Euclidienne normalisée au carré entre deux points haute dimension x_i , x_j :

$$d_{ij}^2 = \frac{\|x_i - x_j\|^2}{2\sigma_i^2} \qquad (2)$$

Dans l'espace à faible dimension, les éléments de la distribution des similarités de Q sont calculés comme suit:

$$q_{ij} = \frac{e(-\|y_i - y_j\|^2)}{\sum_{k \neq i} e(-\|y_i - y_k\|^2)} \qquad (3)$$

Le but de l'intégration est de faire correspondre ces deux distributions autant que possible.

Ceci est réalisé en minimisant une fonction de coût qui est une somme des divergences de Kullback-Leibler [74] entre les distributions originales (p_{ij}) et induites (q_{ij}) sur les voisins pour chaque donnée:

$$C = \sum_i \sum_j p_{ij} \ log \frac{p_{ij}}{q_{ij}} \qquad (4)$$

6. Calcul de la similarité et extraction du résumé:

Après la réduction de dimensions de l'espace, on passe à l'étape suivante qui nécessite le calcul du degré de similarité entre trames.

On pose : $F^{(k)} = \{f_1, f_2, ..., f_{Nk}\}$ l'ensemble des trames de k-ième vidéo.

Notre travail consiste à patronner tous les trames pour les classer par la suite, en se basant sur le principe de subspace clustring en calculant les similarités intra-vue et inter-vue.

Nous allons diviser l'ensemble $F^{(k)}$ en différents « paquets ou sous-groupes » homogènes, en ce sens que les trames de chaque sous-groupes partagent des caractéristiques communes c'est-à-dire le degré de similarité élevé, que l'on définit en introduisant des mesures de distance entre trames.

La similarité est un critère important pour l'identification de sous-groupe dans un groupe d'objets, dans un « espace » ou système.

6.1 Similarité intra-vue :

Nous allons utiliser la distance euclidienne pour calculer le degré de similarité entre deux trames comme suit :

Soit : - $x_i^{(k)}$ le vecteur caractéristique de i-ième trame de la k-ième vidéo ;

- $x_j^{(k)}$ le vecteur caractéristique de j-ième trame de la k-ième vidéo ;

Tel que : $x_i^{(k)} = \{a_1, a_2, ..., a_m\}$ où m = 100 la taille du vecteur caractéristique ;

$x_j^{(k)} = \{b_1, b_2, ..., b_m\}$;

La distance 'd' entre ces deux trames est données par la formule suivante :

$$d(x_i^{(k)}, x_j^{(k)}) = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + \cdots + (b_m - a_m)^2} \qquad (5)$$

On obtient une matrice qu'on va la nommé une matrice de similarité mais qu'elle n'est pas normalisé. Pour la rendre normalisé on doit la diviser par σ^2 tel que σ est l'écart type.

$$\sigma = \beta . \max(d_{eucld}) \qquad (6)$$

Où β c'est un facteur qui a pour valeur : $\beta \leq 0.2$ selon la référence [31]

Donc la distance euclidienne normalisée est :

$$d(x_i^{(k)}, x_j^{(k)}) = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + \cdots + (b_m - a_m)^2}\,/\sigma^2 \quad (7)$$

Les trames vidéo sont classifiées dans des sous-groupes. Chaque fois que la distance d'une trame par rapport aux groupes existants est supérieure à un certain seuil 'S' un nouveau sous-groupe est créé.

Le seuil S est calculé comme suit : $S = 1 - \sigma$ (8)

Ce processus se répète tout en long des séquences de la même vidéo qui garantit la corrélation Intra-vidéo.

6.2 Similarité inter-vue :

Pour calculer la similarité inter-vue on va appliquer l'algorithme de de Scott et Longuet-Higgins. On va appliquer cet algorithme par rapport à chaque deux sous-groupe de deux différentes vue.

- L'approche consiste à calculer dans un premier temps une matrice de proximité G entre une vue i et j, où :

$$G_{ij} = e(\frac{-d_{ij}^2}{2\sigma^2}) \quad (9)$$

  Où d_{ij} est la distance Euclidienne entre deux points u_i et v_j ;

- On réalise ensuite une décomposition en valeurs singulières (SVD) de la matrice G :

$$G = VDU^T \quad (10)$$

- puis on définit une nouvelle matrice P en remplaçant toute les valeurs diagonales de D par 1 : $P = VEU^T$ où E est la matrice diagonale telle que : $E_{ii} = 1$
- enfin, u_i et v_j est considéré comme similaire, si l'élément P_{ij} de P est le plus grand élément de la ligne i et de la colonne j
- l'algorithme sera répéter tout en long de toutes les vues jusqu'à l'obtention des groupes homogène en terme de similarité, comme le montre la figure 59.

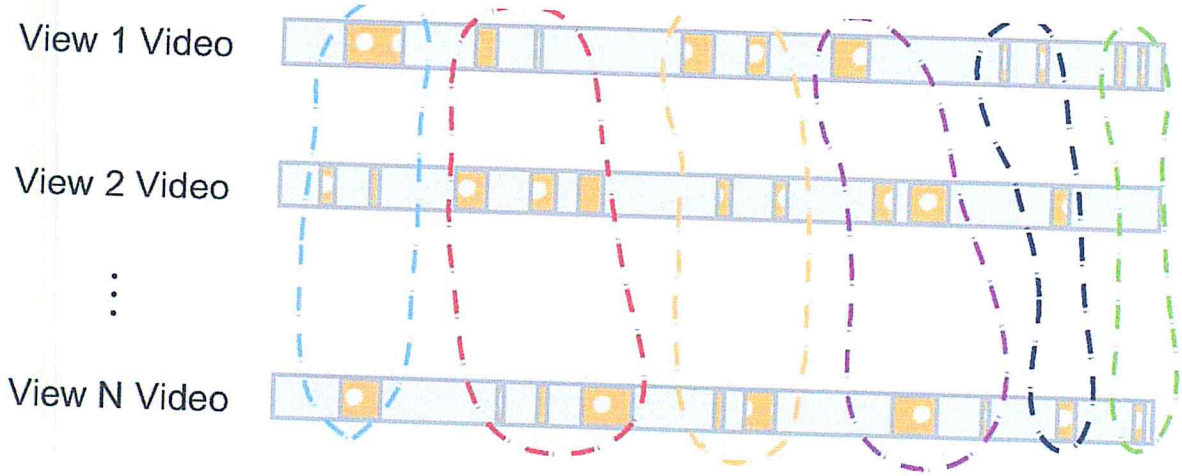


Fig 59: le regroupement des trames similaire

7. création du résumé vidéo :

Après le regroupement de tous les sous-groupes, on essaye d'extraire les key-frame par rapport à chaque sub-space de telle façon qu'un sub-space sera représenter par une trame d'un choix aléatoire avec d'autre trames aux alentour environs de 20 secondes.

Conclusion :

Dans ce chapitre, nous avons détaillé notre approche de construction de résumés vidéo, basée sur l'apprentissage profond en utilisant le modèle BVLC.

Nous avons utilisé la représentation des images sous forme de vecteurs caractéristiques. De même, pour le calcul de la similarité on a utilisé la distance euclidienne, et le principe de subspace clustring pour la classification des trames.

Chapitre V

Présentation des résultats

Introduction :

Après avoir défini tous les concepts liés aux résumés vidéo et défini notre approche de génération de résumé vidéo, nous allons passer maintenant à l'expérimentation.

Nous allons d'abord décrire l'environnement matériel et logiciel utilisé pour notre travail.

Nous décrirons par la suite le jeu de test sur lequel on a travaillé et présenterons les mesures de test utilisées. Enfin nous présenterons les résultats obtenus et allons les discuter.

1- L'environnement matériel :

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

- Marque : HP
- Processeur :Intel® Core™ i7-7500 CPU @ 2.70GHz 2.90 GHz
- Carte graphique :Intel ® HD Graphics 620 et NVIDIA GeForce 930 MX
- Mémoire : 8.00 Go
- Système d'exploitation :Windows 10
- Type du système : Système d'exploitation 64 bit, processeur x64
- CUDA : 8.0

2- L'environnement logiciel:

2.1 Python :

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par Guido van Rossum, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au Centrum voor Wiskunde en Informatica aux Pays-Bas, jusqu'à se voir associer une organisation la « Python Software Foundation », créée en 2001.

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Dès l'instant où nous l'installons sur notre ordinateur, nous disposons de nombreuses fonctionnalités intégrées au langage.[72]

Il existe ce qu'on appelle des bibliothèques qui peuvent être installé et aident le développeur à travailler sur des projets particuliers.

Quelques fonctionnalités du Python :[72]

- de petits programmes très simples, appelés scripts, chargés d'une mission très précise sur notre ordinateur ;

- des programmes complets, comme des jeux, des suites bureautiques, des logiciels multimédias, des clients de messagerie
- des projets très complexes, comme des progiciels (ensemble de plusieurs logiciels pouvant fonctionner ensemble, principalement utilisés dans le monde professionnel).

Quelques caractéristiques du Python :[72]

- Riche : librairie standard Python (*standard library*, i.e. modules intégrés à la distribution de base) couvrant la plupart des domaines, très nombreux *packages* et *modules* d'extension (calcul scientifiques, visualisation, SGBD, réseau...) ainsi que *frameworks* (web avec Django, jeux avec PyGame...)
- ouvert et multiplateforme/portable : libre et open source (licence PSF GPL-compatible sans restriction copy-left) donc librement utilisable et distribuable, disponible sur tous les OS (intégré d'office sous certains, comme Linux et macOS), applications tournant sur toutes les plateformes (le langage étant interprété)
- facilité d'apprentissage et de mise en oeuvre : syntaxe simple/légère, grande lisibilité, flexibilité et efficacité

2.2 OpenCV :

OpenCV (**Open** Source Computer Vision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, 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.

2.3 Numpy :

Numpy est un est une extension du langage de programmation Python, destinée à la manipulation des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Elle propose des types et des opérations beaucoup plus performants que ceux de la bibliothèque standard.

2.4 Scipy :

SciPy est un projet visant à unifier et fédérer un ensemble de bibliothèques Python à usage scientifique. Scipy utilise les tableaux et matrices du module NumPy. Cette

distribution de modules est destinée à être utilisée avec le langage interprété Python afin de créer un environnement de travail scientifique très similaire à celui offert par Scilab, GNU Octave, Matlab.

Il contient par exemple des modules pour l'optimisation, l'algèbre linéaire, les statistiques, le traitement du signal ou encore le traitement d'images. Il offre également des possibilités avancées de visualisation grâce au module matplotlib.

2.5 CUDA :

CUDA (initialement l'acronyme de Compute Unified Device Architecture) est une technologie de GPGPU (General-Purpose Computing on Graphics Processing Units), c'est-à-dire utilisant un processeur graphique (GPU) pour exécuter des calculs généraux à la place du processeur (CPU). Elle a été développée par Nvidia pour ses cartes graphiques GeForce 8 Series.[73]

2.6 CAFFE :

CAFFE est l'abréviation de « Convolution Architecture For Feature Extraction » c'est un open framework pour le deep learning développé par Berkeley Vision and Learning Center (BVLC), qui offre une bibliothèque open-source et des exemples de travail pour un apprentissage profond.

3- Dataset :

Notre dataset est composé de trois jeux de données vidéo multi-vues qui sont :

- **Compus :** où 19 caméras de surveillance qui sont installées au 7ème étage du bâtiment BerryLam de l'Université Nationale de Taiwan, qui couvre l'ensemble du couloir et l'une des salles de bureaux. Toutes ces vidéos ont une durée de sept minutes et dix secondes (07 :10).
- **Office :** Il s'agit du jeu de données office1, qui a été pris avec 4 caméras. Les quatre vidéos ne sont pas synchronisées. Pour cela les durées sont comme suit :

  Vidéo 1 = 14 :58 ;

  Vidéo 2 = 09 :04 ;

  Vidéo 3 = 11 :22 ;

  Vidéo 4 = 14 :58 ;

- **Lobby :** Il s'agit du jeu de données lobby, qui a été pris avec 3 caméras dans un grand hall d'entrée.

4- Outils de mesure :

Pour 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 trames Pertinentes	Nombre de trames Non Pertinentes	total
Nombre de trames Retrouvé (ou proposé)	a	b	a+b
Nombres de trames Non retrouvé (ou non proposé)	c	d	c+d
total	a+c	b+d	a+b+c+d

Tableau 2: calcul des paramètres Recall, Precision est F-mesure

Où :

Retrouvé : signifie que les trames existent dans le résumé proposé.

Pertinente : signifie que les trames existent dans le résumé généré (créé).

➢ **Rappel (recall) :** Rappel exact par rapport à l'ensemble de trames retrouvées. Le rappel mesure la capacité du système à restituer l'ensemble de trames pertinentes. [64].

$$\textbf{Recall} = \frac{\textbf{nombre de trames pertinentes retrouvé}}{\textbf{nombre de trames pertinentes}} = \frac{a}{a+c} \quad [41]$$

➢ **Precision:** Mesure la capacité du système à ne restituer que des trames pertinentes [71]

$$\textbf{Precision} = \frac{\textbf{nombre de trames pertinentes retrouvé}}{\textbf{nombre de trames retrouvés}} = \frac{a}{a+b} \quad [41]$$

➢ **F-mesure :** Mesure qui combine le rappel et la précision. En effet, le rappel et la précision ont tendance à varier en sens inverse. [64]

$$\textbf{F-mesure} = 2 . \frac{Recall * Precision}{Recall + Precision} \quad [64]$$

5- Résultat et discussion :

Méthodes	Office			Compus			Lobby		
	P	R	F	P	R	F	P	R	F
Random Walk	100	61	76.19	70	55	61.56	100	77	86.81
Rough Sets	100	61	76.19	69	57	62.14	97	74	84.17
Notre approche	100	73	84.48	84	69	75.42	100	79	88.26

Tableau 3 : comparaison de performance par rapport aux différentes approches

où P = Precision R = Recall F = F-mesure

Tableau 3 montre les résultats de résumé vidéo sur les trois jeux de données multi-vues « Office, Compus et Lobby ».

Pour les deux ensembles de jeu de données « Office et Lobby », notre approche produit des résumés avec la même précision que RandowWalk.

Cependant, l'amélioration de la valeur de rappel « Recall » environ de 12% pour Office et environ de 2% 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 9% pour Office et environ de 2% pour Lobby par rapport à Random Walk et de 4% par rapport à Rough Sets.

Dans l'ensemble, sur tous les ensembles de données, notre approche est supérieure à toutes les lignes en termes de F-measure.

On voit bien que l'utilisation d'un module d'apprentissage profond pour l'extraction des caractéristique des images et la méthode de subspace clustering pour la classification et la sélection des trames clé et beaucoup mieux que la méthode basé sur la méthode de graphe des shots spatio-temporel (Random Walk) et la méthode basé sur les cartes de corrélation à image clé.

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 compus ».

Conclusion générale

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 et importance ; il permet d'extraire des informations utiles et récapitulatives ce que nous fait gagner un temps considérable.

Notre travail consistait à trouver les images clés qui peuvent résumer toutes les vidéos en supprimant les informations inutiles et redondantes.

Nous avons proposé une méthode pour la génération du résumé vidéo multi-vue qui se base sur l'apprentissage profond. Nous avons utilisé l'architecture neuronales BVLC à base de réseaux de neurones convolutifs afin d'extraire toutes les caractéristiques profondes de la vidéo. Ensuite nous avons employé le mécanisme du subspace clustering pour faire la classification et l'extraction des images clé et générer notre résumé statique.

On a remarqué que notre approche donnait de meilleurs résultats, mais toutefois on a constaté que le temps d'exécution est long.

Perspectives:

Bien qu'on ait aboutit à de bons résultats, le travail peut être amélioré :

- L'utilisation d'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 est envisageable.
- La réduction du temps d'extraction de caractéristiques est importante pour que la solution soit exploitable.

Bibliographie

[1] Marion. A. Introduction aux techniques de traitement d'images, Livre sur le traitement d'images. Edition Eyrolles, 1987.

[2] R.C. Gonzales et P. Wintz. Digital Image Processing, Livre sur le traitement d'image. Edition Wessley 1997.

[3] Nuno Vasconcelos and Andrew Lippman. A spatiotemporal motion model for video summarization. IEEE Computer Society Conference on Computer Vision and Pattern Recognnition, pages 361—366, 23-25 June 1998.

[4] A. Pope, R. Kumar, H. Sawhney, and C. Wan. Video abstraction: summarizing video content for retrieval and visualization. Conference Record of the Thirty-Second Asilomar Conference, I:915—919, 1998.

[5] Alan Hanjalic and Hong Jiang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE Transactions on circuits and systems for video technology, 9(8):1280—1288. December 1999.

[6] Andreas Girgensohn and John Boreczky. Time-constrained keyframe selection technique. IEEE International Conference on Multimedia Computing and Systems, I:756—761.1999.

[7] Shingo Uchihachi and Jonathan Foote. Summarizing video using a shot importance measure and a frame-packing algorithm. IEEE International Conference on Acoustics, Speech, and Signal Processing, VI:3041—3044, 1999.

[8] H. Ueda, T. Miyatake, and S. Yoshizawa. An interactive natural motion picture dedicated multimedia authoring system. ACM SIGCHI 91, pages 343—350, 1999.

[9] D. Zhong, H.J. Zhang, and S.F. Chang. Clustering methods for video browsing and annotation. SPIE on Storage and retrieval from image and video databases, IV:239-246, 1999.

[10] Minerva M. Yeung and Boon-Lock. Time-constrained clustering for segmentation of video into story units. 13th International Conference on Pattern Recognition, III:375-38O, 25-29 Aug 1999.

[11] Y. Zhuang, Y. Rui, T.S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. IEEE International Conference on Image processing. ICIP'98, I:866—870, 4-7 October 1999.

[12] V.Di Lecce, G.Dimauro, A.Guerriero, S.Impedovo, G.Pirlo, and A.Salzo. Image basic features indexing techniques for video skimming. IEEE International Conference on Image Analysis and Processing, pages 715—720, 27-29 September 1999.

[13] Jeho Nam and Ahmed H. Tewfik. Video abstract of video. IEEE 3rd Workshop on Multimedia Signal Processing, pages 117—122, 13-15 September 1999.

[14] N. D. Doulamis, A. D. Doulamis, A. D. Avrithis, and S. D. Kollias. A stochastic framework for optimal key frame extraction from mpeg video databases. Computer Vision and Image Understanding. Special issue on content-based access for image and video libraries, 75:3—24, July/August 1999.

[15] M. Schuster et K.K. Paliwal : Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681, 1999.

[16] H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu. Video parsing, retrieval and browsing: An integrated and content-based solution. ACM International Conference on Multimedia, pages 15—24, 1999.

[17] Par le groupe Adobe Dynamic Media. Initiation à la vidéo Numérique. 2000

[18] N. D. Doulamis, A. D. Doulamis, A. D. Avrithis, and S. D. Kollias. A stochastic framework for optimal key frame extraction from mpeg video databases. Computer Vision and Image Understanding. Special issue on content-based access for image and video libraries, 75:3—24, July/August 2000.

[19] Patrizio Campisi and Alessandro Neri. Synthetic summaries of video sequences using a multiresolution based key frame selection technique in a perceptually uniform color space. International Conference on Image Processing, II:299—302, 2000.

[20] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakate. Videomap and videospaceicon: Tools for anatomizing video content. ACM INTERCHI'93, pages 131—141, 2000.

[21] B. Günsel, Y. Fu, and A.M. Tekalp. Hierarchical temporal vidéo segmentation and content characterization. SPIE on Mulimedia Storage and Archiving Systems II, 3229:46—56, 2000.

[22] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Scene determination based on video and audio features. IEEE Conference on Multimedia Computing and Systems, pages 07—11, 2000.

[23] Y. Taniguchi, A. Akutsu, and Y. Tanomura. Panaramaexcerpts: Extracting and packing panoramas for video browsing. ACM International Conference on Multimedia, pages 427—436, 2000.

[24] A. Mufit Ferman and A. Murat Tekalp. Multiscale content extraction and representation for video indexing. SPIE on Multimedia Storage and Archiving Systems II, 3229:23-31, 2000.

[25] Frederic Dufaux. Key frame selection to represent a video. International Conference on Image Processing, II:275—278, 2000.

[26] Yihong Gong and Xin Liu. Video summarization using singular value decomposition. IEEE International Conference on Computer Vision and Pattern Recognition, II:174-180, 13-15 June 2000.

[27] Yihong Gong and Xin Liu. Generating optimal video summaries. IEEE International Conference on Multimedia and Expo, III:1559—1562, 30 July-2 August 2000.

[28] H. Martin and R. Lozano. Dynamic video abstract generation using an object dbms. IEEE International Conference on Multimedia and Expo, 3:1523—1526, 2000.

[29] Patrick Chiu, Andreas Girgensohn, Wolf Polak, Eleanor Rieffel, and Lynn Wilcox. A genetic algorithm for video segmentation and summarization. IEEE International Conference on Multimedia and Expo. ICME2000, III:1329—1332, 2000.

[30] A. Stefanidis, A. Partsinevelos, and A. Doucette. Summarizing video datasets in the spatiotemporal domain. 11th International Workshop on Database and Expert Systems Applications, pages 906—912, 4-8 September 2000.

[31] Ref : J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI, vol. 22, no. 8, pp. 885–905, 2000.

[32] Sarah V Porter, Majid Mirmehdi, and Barry T Thomas. Detection and classi_cation of shot transitions. In BMVC, pages 1_10, 2001.

[33] Hyun Sung Chang, Sanghoon Sull, and Sang Uk Lee. Efficient video indexing scheme for content-based retrieval. IEEE Transactions on Circuits and Systems for Video Technology, 9(8):1269—1279, December 2002.

[34] Janko Calic and Ebroul Izquierdo. Efficient key-frame extraction and video analysis. IEEE International Conference on Information Technology: Coding and Computing (ITCC'02), pages 28—33, 8-10 April 2002.

[35] Allan Hanbury. Morphologie Mathématique sur le Cercle Unité, avec applications aux teintes et aux textures orientées. PhD thesis, École Nationale Supérieure des Mines de Paris, 2002.

[36] Itheri Yahiaoui. Construction automatique de résumés videos, Proposition d'une méthode générique d'évaluation. 2003

[37] Keesook J. Han and Ahmed H. Tewfik. Eigen-image based video segmentation and indexing. IEEE International Conference on Image Processing, II:538—541, 26-29 October 2004.

[38] Fernando Santos Osorio. Inss : un système hybride neuro-symbolique pour l'apprentissage automatique constructif. 2004

[39] Fernando Santos Osorio. Inss : un système hybride neuro-symbolique pour l'apprentissage automatique constructif. 2004

[40] Cécile Kattnig. Gestion et di_usion d'un fond d'image. Armand Colin, 2005.

[41] NAKACHE Didier, METAIS Elisabeth. Evaluation: nouvelle approche avec juge. 2005

[42] Mickael Guironnet. Méthodes de résumé de vidéo à partir D'informations bas niveau, du Mouvement de camera ou de l'attention Visuelle. 2007

[43] Mickael Guironnet. Méthodes de résumé de vidéo à partir D'informations bas niveau, du Mouvement de camera ou de l'attention Visuelle. 2007

[44] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, Speededup robust features (surf), Comput. Vis. Image Underst., vol. 110, no. 3, pp. 346–359, June 2008.

[45] Anne-Claire MAHEO L'université Jaume I 12071 Castellón de la plaña Espagne. Méthodes de suivi d'un objet en mouvement sur une vidéo. 2009

[46] Yoshua Bengio. Learning Deep Architectures for AI, Foundations and Trends in Machine Learning, 2(1). 2009

[47] M.Calonder, V. Lepetit, C. Strecha, and P. Fua, Brief: Binary robust independent elementary feature, . In *In European Conference on Computer Vision*, 2010.

[48] Dominik Francoeur. Machines à vecteurs de support Une Introduction. 2010

[49] Yanwei Fu and al. Multi-View Video Summarization. 2010

[50] Nabeel Younus Khan, Brendan McCane, and Geoff Wyvill, SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset, International Conference on Digital Image Computing: Techniques and Applications, pp.501-506, 2011.

[51] Ping Li, Yanwen Guo, Hanqiu Sun. Multi-keyframe abstraction from videos. 2011

[52] Kathleen Mullani_. Fade in fade out. 2012.

[53] Nicolas LABROCHE. Méthodes d'apprentissage automatique pour l'analyse des interactions utilisateurs. 2012

[54] T.J. Haykin, Neural network, a comprehensive foundation, Prentice-Hall, 2012.

[55] P M Panchal, S R Panchal, S K Shah, A Comparison of SIFT and SURF, International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013.

[56] Faicel CHAMROUKHI. Classification supervisée : Les K-plus proches voisins. 2013

[57] Moez BACCOUCHE. Apprentissage neuronal de caractéristiques spatio-temporelles pour la classification automatique de séquences vidéo. 2013

[58] M. Wybier et P. Bossard, "Musculoskeletal imaging in progress : the eos imaging system", *Joint Bone Spine*, vol. 80, no. 3, pp. 238–243, 2013.

[59] Matthew D. Zeiler, Rob Fergus. Visualizing and Understanding Convolutional Networks. 2013

[60] Gilles Deleuze. Cinéma 1-L'image-mouvement. Minuit, 2014.

[61] Alexandra Derntl, Survey of Feature Detectors and Descriptors in Surgical Domain, 2014.

[62] Koudri Mohammed. Apprentissage automatique. 2014

[63] William Thong. Apprentissage de représentations pour la classification d'images biomédicales. 2015

[64] Alain Baccini. Analyse des critères d'évaluation des systèmes de recherche d'information. 2015

[65] Nikita Kaushik1, Ritu Rawat2, Anshika Bhalla3. A Brief Study of Different Feature Detector and Descriptor. April 2016

[66] Pami Prakash, Anil A.R, Survey on Key Feature Descriptors Used In Computer Vision Applications, International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-1, January 2016.

[67] Abir Das, Amit K. Video Summarization in a Multi-View Camera Network. 2016

[68] Mostefaui Souad, Aichouch Hadjer. Génération des résumés de vidéo. 2015-2016

[69] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets

[70] http:// blogs.msdn.microsoft.com/big_data_france/l'apprentissage automatique

[71] http://adeshpande3.github.io/adeshapande.github.io/the 9 deep learning papers you need to know about.html

[72] http://enacit1.epfl.ch/introduction.python

[73] http://fr.wikipedia.org/wiki/compute_unified_devise_architecture

[74] Geoffrey Hinton and Sam Roweis. Stochastic Neighbor Embedding